ABSTRACT

Title of dissertation: RICH AND SCALABLE MODELS FOR TEXT

Thang Dai Nguyen, Doctor of Philosophy, 2019

Dissertation directed by: Professor Jordan Boyd-Graber
Department of Computer Science and
Institute for Advanced Computer Studies

Professor Philip Resnik
Department of Linguistics and
Institute for Advanced Computer Studies

Topic models have become essential tools for uncovering hidden structures in big data. However, the most popular topic model algorithm—Latent Dirichlet Allocation (LDA)— and its extensions suffer from sluggish performance on big datasets. Recently, the machine learning community has attacked this problem using spectral learning approaches such as the moment method with tensor decomposition or matrix factorization. The anchor word algorithm by Arora et al. [2013] has emerged as a more efficient approach to solve a large class of topic modeling problems. The anchor word algorithm is high-speed, and it has a provable theoretical guarantee: it will converge to a global solution given enough number of documents. In this thesis, we present a series of spectral models based on the anchor word algorithm to serve a broader class of datasets and to provide more abundant and more flexible modeling capacity.

First, we improve the anchor word algorithm by incorporating various rich priors in the form of appropriate regularization terms. Our new regularized anchor word algorithms

produce higher topic quality and provide flexibility to incorporate informed priors, creating the ability to discover topics more suited for external knowledge.

Second, we enrich the anchor word algorithm with metadata-based word representation for labeled datasets. Our new supervised anchor word algorithm runs very fast and predicts better than supervised topic models such as Supervised LDA on three sentiment datasets. Also, sentiment anchor words, which play a vital role in generating sentiment topics, provide cues to understand sentiment datasets better than unsupervised topic models.

Lastly, we examine ALTO, an active learning framework with a static topic overview, and investigate the usability of supervised topic models for active learning. We develop a new, dynamic, active learning framework that combines the concept of informativeness and representativeness of documents using dynamically updating topics from our fast supervised anchor word algorithm. Experiments using three multi-class datasets show that our new framework consistently improves classification accuracy over ALTO.

RICH AND SCALABLE MODELS FOR TEXT

by

Thang Dai Nguyen

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2019

Advisory Committee:
Professor Jordan Boyd-Graber, Chair/Co-Advisor
Professor Philip Resnik, Co-Advisor
Professor Douglas W. Oard
Professor Naomi Feldman
Professor Furong Huang

Acknowledgments

I owe my deepest gratitude to many wonderful people, whose help and support have made this dissertation possible. Spending seven years at the University of Maryland part-time while working full time is a very challenging experience, but it also shows that I have been fortunate to reach this milestone in my life. This would not be possible without the support and guidance from the following people.

First and foremost, I wholeheartedly thank my co-advisors, Jordan Boyd-Graber and Philip Resnik, for their continuous guidance and advice throughout my graduate study. They have been a fantastic team that provides so much support for me as I needed; their beautiful lectures have built my foundation for research; their wisdom helped me cross the boundary to graduation. In particular, I would like to thank Jordan for accepting me as a graduate student at the ISchool; without that first step, I could not have been transferred to become a graduate student at the Department of Computer Science at the University of Maryland. Jordan has been patiently laying the bricks for my research, guiding me from the first ACL conference talk to the last research equation. I am deeply grateful to Philip for guiding me through hardship; every time I was stuck with research, either emotionally or technically, Philip was the first one I came for advice. I also miss our lunches at the Lilit Cafe a lot.

I am fortunate to know many amazing professors whose research have been inspirational for my dissertation. I would like to thank Professor Hal Daumé III, Professor Jimmy Lin, Professor Thomas Goldstein, Professor Kevin Seppi, and Professor Eric Ringger for intellectual conversations. My special thank to Professor Neomi Feldman, Professor Douglass W.Oard, and Professor Furong Huang for agreeing to serve on my thesis committee and spending their precious time reading my dissertation.

Many thanks also go to wonderful friends who have coauthored and collaborated with me on various papers. I thank Yuening Hu, Jeff Lund, William Armstrong, Leonardo Claudino for working with me on experiments and on writing and for continually providing feedbacks and insights.

I am fortunate to be part of a fantastic community of researchers: the Computational Linguistics and Information Processing Lab (**CLIP**). Various ideas and inspiration come from discussions with Mohit Iyyer, Alvin Grissom II, Viet-An Nguyen, Mossaab Bagdouri, He He, Ke Zhai, Khanh Nguyen, Ning Gao, Jyothi Vinjumur, Jinfeng Rao, Sudha Rao, Weiwei Yang, and Snigdha Chaturvedi. I also want to especially thank many UMIACS staffs for their help and the Computer Science professors for their fascinating lectures.

As a part-time student, my Ph.D. is not possible without constant support from my managers at work. I especially thank Dr. Olivier Bodenreider and Lee Peters at the National Library of Medicine (NLM) for their continuous trust and encouragement so that I could take time off work to go to classes while still progressed and performed well at work.

Finally, I want to thank my family for their love and support. I thank my parents for their unconditional love; they always believe in me and encourage me to pursue my dreams. I thank my sister for her advice and encouragement. And most importantly, I thank my beloved wife Hang and our beautiful sons, Thomas and Ryan. My wife has always been wonderful, being there whenever I am high or low in this adventure; my sons have been a big push for me to move forward. I know, together, we could accomplish anything in life.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Statistical Machine Learning in The Age of Big Data

We are living in an era of vast information. Every day we produce and consume data through multiple sources, ranging from devices such as phones, cameras, computers, and cars to activities such as doing research, conducting surveys, or using social media websites. The Internet has become the world's main communication channel because of cheap and powerful computers, networks, and storage devices.

This massive amount of data also comes in various types and shapes—from pure text messages to long written documents, from images to videos, from complex multi-dimensional data satellite images to DNA sequences and time series datasets such as those used in weather simulation [Hearst, 1999, Deng et al., 2009, Walker, 2014, Jean et al., 2016, Libbrecht and Noble, 2015, Xu et al., 2015] and drug forecasting [Cook, 2015]. Many current datasets are enormous not only in volume but also in the complexity of relationships among data points [Leskovec and Krevl, 2014, Bollacker et al., 2008] and communities [Yang and Leskovec, 2015].

Collecting massive amounts of data comes with the need to analyze them. Having powerful analysis tools will accelerate our understanding of data. Also, better data analysis tools will make us better able to draw meaning out of data and will allow us to improve the ways we work with data [Manyika et al., 2011, Cukier, 2014, Norvig, 2011]. The more we understand, the more accurate the decisions we can make.

The availability of data and the need to analyze them provide researchers opportunities to contribute through the research and development of powerful analysis tools and algorithms. For example, public collections of emails allow researchers to build spam filtering algorithms to understand the behaviors of spammers and to better filter out malicious emails. Social media posts such as tweets allow us to study psychological issues such as depression [Cavazos-Rehg et al., 2016, Tsugawa et al., 2015, Resnik et al., 2015], to study social effects in the face of political changes [Grimmer, 2015, Wong et al., 2016, Beauchamp, 2016], to predict human traits [Orehek and Human, 2017], and to uncover sentiments and opinions [Severyn and Moschitti, 2015, Rosenthal et al., 2013, 2015, Mohammad et al., 2016, Balikas and Amini, 2016]. Vast collections of images and captions (e.g., YouTube) allow us to understand and build intelligent algorithms for image segmentation [Chen et al., 2018, Badrinarayanan et al., 2015, Long et al., 2015] and object recognition [Abu-El-Haija et al., 2016, Chen et al., 2015b, Zhou et al., 2014, Lowe, 1999, Duygulu et al., 2002].

Our understanding of voluminous data comes mostly from the application of statistical models that learn and extract useful relationships. In the context of machine learning models, "features" or "variables" capture the statistical properties of data [Hastie et al., 2009, Guyon and Elisseeff, 2003]. Correlations among these variables allow us to make inferences, giving understanding for tasks such as making predictions or building business rules. In the online product market, for example, comments from consumers have been used by supervised learning systems to make better predictions and to provide more insight into customers' buying behaviors [Liu, 2012]. Social and political activities have never been better understood, given the help of statistical analysis and machine learning tools [Nguyen et al., 2015b]. Weather prediction [SHI et al., 2015, Sharma et al., 2011], autonomous cars [Chen et al., 2015a, Teichmann et al., 2018], and spam detection are a few examples where we have seen great successes in the application of advanced statistical models to large datasets.

Scalability: The Curse of Dimensionality

Yet, working with vast and complex data is very challenging; most models eventually reach their performance plateau and cannot scale well to bigger datasets. Solving most problems becomes extremely laborious as datasets grow bigger. The bottleneck of training large and complex models is prohibitive to such a degree that some models are no longer useful when they have to deal with a big dataset. As data grow, we tend to use more variables and more complex models to capture variance in those data, and yet we still hope to be able to generalize the models so that they will perform well on the future, unseen datasets. But working with millions of variables requires a different perspective and more in-depth thinking than working with just a few hundred variables. We work within the boundaries of time and computational power, and, while striving for scalability, we often have to fight against the *curse of dimensionality* [Bellman, 1957, Marimont and shapiro, 1979, Chazelle, 1994, Chávez et al., 2001]. The *curse of dimensionality* is this:

> When the dimensionality increases, the volume of the space increases so fast (exponentially) that the available data become sparse. This sparsity is problematic for any method that requires statistical significance. In order to

obtain a statistically sound and reliable result, the amount of data needed to support the result often grows exponentially with the dimensionality.

Despite the curse, improving the scalability of statistical machine learning models is critical for understanding large datasets and for supporting the process of making accurate and timely decisions.

Variability: Data Heterogeneity

Another issue that impacts how we build statistical machine learning models is that data may come from multiple sources. To solve a specific business problem in a particular domain, we often have to prepare datasets from many different sources; sometimes the biggest task is to combine these large siloed datasets, both structured and unstructured, into a single source. In addition to this, we also need a smart, efficient way to incorporate all sorts of related information, such as raw data, metadata, domain ontologies, and expert advice.

After we have somewhat successfully prepared a set of data sources and consolidated them, we then need to either apply standard statistical approaches or devise new approaches tailored to the characteristics of the particular dataset. In biomedical domains, for example, many biomedical ontologies have been incorporated into supervised and unsupervised learning methods to better capture information [Chang et al., 2018, Mamoshina et al., 2016] or to do cell segmentation and prediction [Ronneberger et al., 2015]. Finally, in many fields a large amount of expert knowledge is available in various forms, such as documents, reports, presentations, and talks, for smart machine learning models to take advantage of, thereby improving upon current models [Xu et al., 2014, Hu et al., 2009, Andrzejewski et al., 2009b, 2011, Jagarlamudi et al., 2012, Xie et al., 2015a].

Interactivity: Interaction with Humans

Statistical machine learning models work as good as the quality of the datasets fed to them. In other words, if the quality of the datasets is poor, statistical models will suffer. This challenge can be overcome by involving humans in the loop. Instead of relying one hundred percent on statistical algorithms, we could find ways to insert humans into the system to annotate data and refine models. In practice, the need to involve human users emerges with datasets both large and small. The rule of thumb is that, to avoid significant bias, statistical models require a sufficient number of data points to learn, so it is always better to acquire more data points if possible, and acquiring more useful, high-quality data points requires informed human input.

A well-known example is in the field of *active learning*, where we focus on improving the process of label collection (annotation) by assuming that supervised learning models can learn with a smaller number of data points if those data points are of higher quality. This motivates us to query users for the best possible labeled data points to build a training set.

There is a two-fold advantage to this approach. First, querying users for quality labels reduces the cost of amassing sizeable labeled training sets. Second, active learning still ensures a high prediction accuracy for the supervised learning models because models play a role in the labeling process. Active learning has found numerous applications in many practical domains, especially in domains where acquiring labels is particularly costly [Settles, 2010, Wang and Hua, 2011, Hoi et al., 2006, Tong and Koller, 2002].

Another intriguing example where we involve humans in machine learning is where we find users themselves to be part of the learning cycle. In the typical example of *interactive machine learning*, where there is a need to ingest a large amount of data, complex machine learning algorithms do not always produce results that accurately capture what users need, so we have users work directly with algorithms to refine their outputs [Fails and Olsen, 2003, Settles, 2011a]. This approach is especially useful for working with textual content such as document summarization or topic modeling [Liu et al., 2009, Eisenstein et al., 2012, Hu et al., 2014a, Hoque and Carenini, 2015]. While active learning frameworks are applicable only for supervised learning, interactive machine learning frameworks are useful for both supervised and unsupervised learning problems. However, because both active learning and interactive machine learning frameworks include users in the loop, the essential requirement is the ability to quickly and interactively adjust models as they incorporate user feedback. In the end, speed is essential if the machine learning model is to include users, in the same way that scalability is essential if the machine learning model is to handle large datasets.

Hence, we can summarize the ideal properties of modern statistical machine learning models applied to big data: first, models need to scale well for big datasets; second, models must be rich and flexible to address cases of significant data variability; and third, models need to run fast enough to allow interactivity. In addition, these models have to pass practical downstream evaluation metrics such as prediction accuracy (supervised learning) or model generalization. Figure 1.1 demonstrates three key properties that drive our needs to build better models: scalability, extensibility, and interactivity. In this thesis, we will

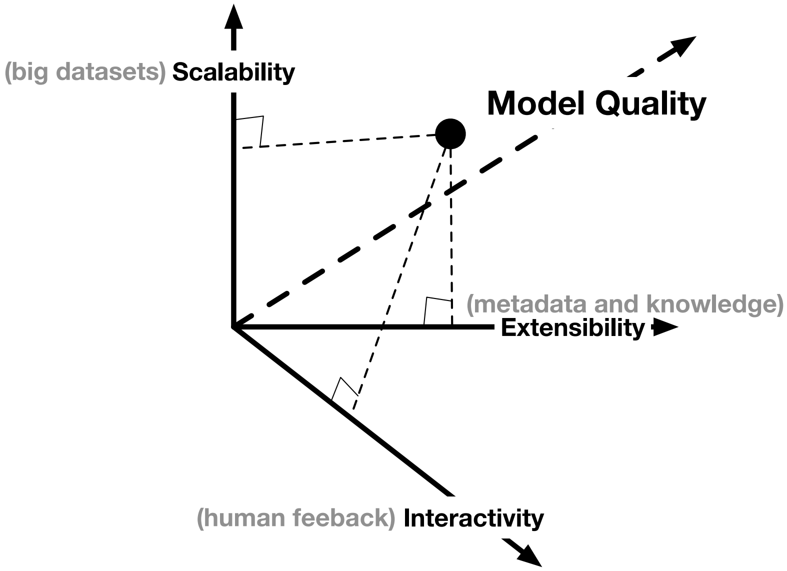


Figure 1.1: A way to evaluate statistical model quality given the current rise of big data. We want a model to handle big datasets (scalability), to be able to address cases of data variability (extensibility), and to run fast enough for humans in the loop (interactivity).

improve these properties by developing a series of models to work with text documents.

1.2 Natural Language Processing

Among all types of data that are available, text is arguably the most popular and pervasive [Hearst, 1999, Joachims, 1998, Dhillon and Modha, 2001, Berger et al., 1996, Socher et al., 2012]. Text provides vast amounts of information and contains a great wealth of knowledge [Alm et al., 2005, Aggarwal and Wang, 2011]. This poses the challenge because analyzing vast amounts of textual data takes time and effort. In addition, text also comes from multiple sources: users' comments from social media and product websites, political corpora such as the Europarl dataset [Hajlaoui et al., 2014] or US Congressional Record, legal documents such as laws and consent forms, student essays, research papers, and text messages. These few examples show that textual content is enormous. Because most textual content comes from everyday natural language usage, it is highly variable and representative of large datasets.

How Language is Different

Natural language is different from other modalities. One interesting characteristic of natural language is how we understand the meanings of its key features—words. Our understanding of a given word comes largely from its relationships to other, surrounding words. In this way it is very different from visual data, for example, in which features are encoded in the image. This difference drives the need for different tools to process and understand natural language.

The high prevalence of natural language text has resulted in a correspondingly high quantity and quality of natural language processing tools that can process and understand text datasets. For example, in the realm of text mining, part-of-speech (POS) tagging and named entity recognition (NER) have found their application in biomedical-term extraction [Settles, 2004], search engines [McCallum and Li, 2003], or social media [Ritter et al., 2011], etc.

Vector Space Model

The vector space model (VSM) is a model for representing textual content such as documents as vectors of words (or terms). VSM was first introduced in the original paper by Salton et al. [1975] for use in a document similarity task. In this original form, each document is a vector of term frequencies (or a column in the term–document matrix). Each document vector has a length equal to the corpus vocabulary size. In this representation, each document is a point in space. Documents are said to be semantically similar if their points are close in this space; they are said to be semantically different if their points are far away in this space. By using VSM, we provide computers the ability to process texts in an automatic way for semantic understanding [Turney and Pantel, 2010].

Another case of VSM that has become more and more popular originates from the task of word similarity. Similar to a document representation, each word representation is also modeled as a vector in space. The dimension of the word space could be the number of documents in the corpus (e.g., with a word vector as a row in the term–document matrix), or the vocabulary size (in the case where we use surrounding words to capture the context of a word), or a fixed, given dimension (in the case of lower-dimensional embedding; see below). In this representation, each word is a point in space. Words are said to be semantically similar if their points are close in this space; they are said to be semantically different if their points are far away in this space.

Because the number of documents varies and is usually large, we often represent words as vectors in lower dimension. Typically, there are several popular techniques to learn word vectors instead of directly using the term–document matrix. Sample methods include matrix factorization using either term–document matrix or word–word co-occurrence statistics [Deerwester et al., 1990, Lebret and Collobert, 2014, Levy and Goldberg, 2014b, Li et al., 2015], neural network models [Mikolov et al., 2013a], and models that explicitly represent words as their context [Levy and Goldberg, 2014a]. The term *word embedding*, which has been used recently to mean word-vector representation, originates with the neural *Word2vec* model by Mikolov et al. [2013a]. Interestingly, Word2vec captures not only word similarities but also a pair of words similarities, such as that the distance between *King* and *man* is equal to the distance between *Queen* and *woman* (and even more interesting, these distance vectors, including magnitude and direction, are roughly equal) . Many subsequent models (not necessarily neural) also show similar properties [Levy and Goldberg, 2014a, Pennington et al., 2014].

Successful deployment of word embedding models in many NLP applications has recently resulted in many black-box tools for generating word embeddings. Examples include Mikolov's Word2vec, Stanford's GloVe, FastText, and Gensim. As we view word embeddings as external knowledge learned from massive corpora such as Wikipedia, we will see in a subsequent chapter that these *black-box* word embeddings can be easily combined with our models.

Distributional Hypothesis

To understand word embedding, we should look back to the seminal work of Latent Semantic Analysis [Deerwester et al., 1990, LSA]. LSA assumes that words will be semantically close if they occur in a similar context, an idea that is called the *distributional hypothesis* in linguistics [Harris, 1945]. The concept of word context is widely applied to many situations and results in many applications and models for learning word embeddings. For example, the context of a given word can be a window of surrounding words [Lund and Burgess, 1996, Mikolov et al., 2013a] or can be defined using grammatical dependencies [Padó and Lapata, 2007].

Big Text Data

Machine learning algorithms have many successful applications to natural language processing. However, due to the complex relationships in languages, such as the word–context relationship, machine learning approaches have to address problems at scale. For example, traditional methods that use a discrete representation for a word struggle with the curse of dimensionality as the vocabulary grows. Modern methods such as VSM-based neural language models fix the sparse representation issues, but the models themselves often have to use a large number of variables. In the subsequent section, we describe several common challenges in applying machine learning models to solve massive NLP problems.

1.3 Challenges of Applying Machine Learning to Natural Language

The above requirements that statistical models be rich and scalable will apply naturally to the solution of many problems of big text data. The key to success is the combination of machine learning and natural language processing techniques. There are several important questions that, if answered, will accelerate research and application of statistical models for text documents:

1. How can we create effective NLP models to understand large-scale collections of text documents?

2. How can we create effective NLP models to learn highly variable types of data and domain knowledge?

3. How can we utilize knowledge from humans to create effective NLP models?

We address these three questions by looking at the three challenges of working with large-scale document collections: scalability, variability, and interactivity.

1.3.1 Scalability Challenge

Scalable machine learning algorithms often resort to using the computing platform (e.g., distributed or parallel); to online learning, where algorithms learn only one instance at a time; or to stochastic techniques (e.g., stochastic gradient descent). However, due to the large number of textual documents and the combinatorial explosion of discrete features within these documents (e.g., words and sentences), it is very challenging to build scalable models for understanding natural language. Simply applying standard techniques often does not produce useful results.

Nevertheless, building automatic understanding systems for natural language is essential, because these systems drive many practical applications. Most successful applications of NLP contain many NLP components that require fast or even real-time processing. For example, a machine translation application often has codes written for named entity recognition, language modeling, or co-reference resolution. Because of these rich inter-relationships within NLP applications (e.g., rich feature interactions inflate the relationship space, causing scalability and sparseness problems), it is a challenge to build them to be fast and scalable.

1.3.2 Variability Challenge

Natural language, such as that present in text documents, has unique issues of variability. Unlike in vision processing, where visual features have fairly consistent semantic meanings (e.g., points, edges, or complex objects such as a cat) that are perceived consistently across viewers, language features such as words are defined in association with the context in which they appear. There are also many languages, and each has its level of complexity—from syntactic representation to semantics and understanding. Even within a single language, different types of documents and corpora also have different vocabularies and different levels of semantic representation. Political language, for instance, is very different from language used in social networks. The enormous variability of language suggests that no single machine learning model will work for all cases.

The variability challenge is compounded by the wide range of domain knowledge that is often stored within text documents. Examples include expert knowledge captured in legal texts (e.g., laws and policies), government regulations, educational materials, or specialized domains such as medical ontologies (e.g., PubMed). Very often, data and knowledge go along with each other; for example, metadata or coding categories are often critical components of reports and surveys about certain topics. To create effective NLP tools and models, we often have to deal with complex relationships of languages within domains, in addition to each domain-specific body of knowledge. The problem becomes more challenging when too little or too much data are available to analyze. In the case of few data points and little or no available knowledge, there are a limited number of patterns and regularities to learn from. This is harder with language due to the sparseness property of languages (e.g., a large vocabulary). In the case of big datasets and messy knowledge bases, we face the challenge of scalability as well as the challenge of coming up with the right NLP models to efficiently capture the relationship of data and knowledge.

Given the large amount of data, the large number of domains, and the variability of text across domains, NLP models have to deal with many uncertain situations. Hence, most NLP models become extraordinarily complex and slow to execute in practice.

1.3.3 Interactivity Challenge

Machine learning algorithms in general, and NLP models specifically, are best used in the supervised learning frameworks where we have substantial labeled data to learn from. Given a large amount of labeled data, most models can do a good job at predicting outcomes of future unseen data points—and this operation can be automatically evaluated using metrics such as accuracy, AUC (area under curve), or ROC (receiver operating characteristic). On the other hand, unsupervised NLP models often use metrics that are not totally accurate or meaningful, and they often require user involvement for evaluation (e.g., to check the quality of generated clusters or topics). Hence, human users are needed in supervised learning cases where very few labeled training data are available; they are also needed in most cases of unsupervised learning. It is also best to involve humans for natural language applications, because humans often provide useful feedback, which can be either controlled feedback (we explicitly ask for their input) or uncontrolled feedback (the record of their activities or interactions with the NLP application). However, looping human users into the system comes with high costs (human resources are expensive), and it requires that we create high-speed NLP models, because humans have a very low tolerance for sluggish systems.

The requirements we need to meet to build an interactive machine learning system for a natural language include not only speed but also extensibility. By extensibility, we mean that after users give feedback through labels, corrections, or rankings, models need to be rich and flexible enough to utilize that information. Given the richness of language, increased interactivity with users motivates machine learning and NLP model designers to continue improving their models. Interactive machine learning is a very new area of research and has many potential breakthrough applications.

1.4 Challenges of Applying Topic Models

While many NLP models achieve various levels of understanding, general latent variable models for languages, such as topic models, stand out as among the most powerful techniques that have achieved practical success. These models assume that certain hidden variables (e.g., explanatory latent variables) could explain observations, and the goal is to identify those latent variables. In topic models, the latent variables are called *topics*, which capture thematic structures from the document collection of interest; inferring topics is a crucial part of understanding those documents.

Topic modeling algorithms have become standard tools for analyzing large collections of text documents. Topic models summarize a corpus using topics, hence offering insights to help us understand unstructured text documents. In a common topic model, each topic is a list of words and is often visualized as the top ten or twenty words that share the context. Unlike popular clustering techniques such as K-means, where a document can belong to one cluster only, in topic models a document may contain several topics. In technical terms, we say that each document is a mixture of various topics. Having this property makes topic models flexible and powerful for understanding many real-world texts because, for example, a news article often includes several topics and themes mixed up together.

The most popular topic model is Latent Dirichlet Allocation (or LDA) by Blei et al. [2003]. LDA learns the posterior estimate of latent variables (topics) given observations (words and documents); in turn, latent variables explain unseen documents or generate new documents. The key difference between LDA and previous models is the use of the sparse Dirichlet distribution as priors for both document-topic and topic-word. This takes into account the property of many real-life documents, in which each document contains only a small number of topics and each topic uses only a small set of words frequently.

The learning step of LDA often involves estimation procedures such as sampling techniques or optimization algorithms. Because it is intractable to learn latent variables directly, estimation procedures often invoke approximation methods. Examples include learning fixed points by Gibbs sampling or by solving optimization problems using coordinate ascent to update optimal variational parameters one at a time. In Gibbs sampling, we sample each variable assuming all other variables are constant. By running many rounds of sampling, Gibbs sampling assumes that the posterior distribution can be found or closely approximated. Variational techniques for LDA work by approximating the posterior distribution with a simpler form, called *variational distribution*, and directly apply optimization techniques to find variational parameters for that variational distribution (see Chapter 2, Section 2.2).

Many versions of LDA have been developed to address various types of text datasets: large-scale text corpora, texts with rich label information, texts with rich relationships, or multi-lingual text documents. These models provide useful topics together with additional outputs that support many downstream tasks such as prediction, visualization, or real-time analysis. However, except for LDA, richer topic models suffer from over-complex modeling issues or from the price of overhead implementations (see Chapter 2, Section 2.4). Many recent efforts have been to scale up and scale out topic modeling algorithms: using the distributed implementation in MapReduce [Zhai et al., 2012], creating online versions [Wang et al., 2011, Wang and Blei, 2012, Bryant and Sudderth, 2012, Li et al., 2014, Hoffman et al., 2013], or creating parallel topic models [Smola and Narayanamurthy, 2010].

Interactive topic modeling is also a new area that has gained recent interest. Examples include using simple interactive visualization for topic models [Eisenstein et al., 2012], directly modeling user feedback using constraints into LDA [Hu et al., 2014a], or building multilingual topic models based on user feedback on alignment across languages [Yuan et al., 2018]. Adding an interactivity layer will create more applications for topic models, since each language domain has its own specific needs. Also, the ability to involve humans into designing topic modeling systems becomes critically important when there are rich domain knowledge and expertise we need topic models to capture.

An area of research in which we see little progress is the theoretical aspects of topic modeling algorithms. Although sampling and statistical theories support probabilistic solutions for topic models, most topic algorithms have only local modes of convergence because they use Gibbs sampling or variational inference approaches (see Chapter 2, Section 2.2). These inference techniques rarely come with any provable theoretical guarantee of convergence: given sufficient sampling size (e.g., a large number of documents), can we guarantee we will find a global solution (e.g., latent topics)? Additionally, the relative accuracy of Gibbs sampling and variational inference is still unknown—understanding these techniques will require new statistical theories [Blei et al., 2016].

Recently, the quest for theoretical properties of topic model solutions has attracted much attention in the machine learning community and people have proposed solutions based on approaches where matrix and tensor decomposition dominate [Anandkumar et al., 2012c,a,b, Arora et al., 2012b, 2013]. The immediate advantage of these techniques is their fast solution, which depends only on word statistics such as second or third moments. These solutions often come from solving a convex optimization and have clear bounds on accuracy (see Chapter 2, Section 2.5.1).

1.5 Contributions to the Anchor Word: Addressing Scalability, Variability, and Interactivity

The anchor word algorithm by Arora et al. [2013] stands out as an efficient solution for topic models such as LDA, because instead of using expensive inference, it uses non-negative matrix factorization (NMF) on the word-by-word co-occurrence information. By assuming that there is a subset of columns that serve as bases so that all other columns are a linear combination of them, we make computing NMF on this co-occurrence matrix straightforward, because the problem becomes convex. In addition, constructing this co-occurrence matrix is done only once; the anchor word algorithm scales with the size of the corpus vocabulary instead of the number of documents in the corpus (the sample size), which is the case in other inference techniques such as Gibbs sampling. We will give a detailed review of the anchor word algorithm in Chapter 2, Section 2.6.

In this thesis, we address in-depth the three critical properties of a good topic modeling algorithm: scalability, extensibility, and interactivity. We directly extend the scalable anchor word algorithm (Section 2.6) to make it more useful and applicable. We argue that scalability is only useful if models perform well on tasks. So even though we take a scalable approach, our approach to improve its quality without changing scalability contributes to scalability itself. We introduce three novel solutions for topic modeling that are based on the anchor word algorithm. Our new topic models are very scalable and fast because they inherit the scalability and speed of the anchor word algorithm.

In our first effort, we significantly improve the quality of topics and increase the robustness of the anchor word algorithm, producing a model that is of higher quality and yet still highly scalable. Our second new model extends this work by improving the extensibility of the anchor word algorithm so that we are able to use it with a broader group of datasets with greater variability. To further address the variability of data, we extend the anchor word algorithm to work with labeled datasets, allowing it to take advantage of side and metadata information (hence creating better topics for domain applications that feature data variability). Our approach is very general because it utilizes distributional hypotheses to model word representation to capture words, metadata, and their relationships, resulting in better predictions. Our final model addresses interactivity by incorporating user input to improve prediction. To do this in an interactive manner, we improve the inference step of the anchor word algorithm. Our approaches combine advantages from probabilistic topic models with the scalable inference scheme of spectral methods to better consume and understand large text datasets. In the remainder of this chapter, we briefly introduce the three models we will present in subsequent chapters. For each model, we give an overview of the problems and applications that the model tries to address and explain how we evaluate it. We believe that having a set of model choices such as ours will benefit researchers as well as organizations that desire to use topic models in practice.

1.5.1 Incorporating Priors into Scalable Anchor Topic Models for Robustness and Extensibility

The anchor word algorithm is fast, however, its topics are lower in quality in comparison to topics produced by probabilistic topic modeling inference schemes such as Gibbs sampling or variational inference. Additionally, the anchor word algorithm is not flexible enough to incorporate external knowledge such as informed priors [Jagarlamudi et al., 2012, Zhai et al., 2012]. Our goal is to create a fast, robust, and high-quality topic model based on the anchor word algorithm. Most current approaches in the literature take a high-quality but slow model (e.g., supervised LDA) and improve its speed and scalability [Zhu et al., 2013]. Our approach moves in the opposite direction: we start with an existing model that is fast and scalable but suffers from low topic quality, and we improve its quality while still keeping its core speed and scalability. We do so by introducing various useful topic priors into the anchor word algorithm. Each prior type will contribute to the robustness and stability of the learning model. Specifically, the Gaussian prior provides the flexibility to add metadata or to inject prior knowledge (e.g., a knowledge graph) by changing the Gaussian mean. We experiment with the Linguistic Inquiry and Word Count (LIWC) software as a knowledge source and change the mean of the Gaussian prior to inject this knowledge into the anchor word algorithm. Being guided by LIWC, our model can find related words more often than a model not being guided by LIWC.

We also introduce a Dirichlet prior over topics into the anchor word algorithm. This idea is similar to many approaches from probabilistic topic models. We show that this approach helps the anchor word algorithm to produce higher-quality topics than the case of no prior—these topics are even on par with high-quality topics produced by Gibbs sampling or variational inference. Given the stability, extensibility, and quality of the proposed models, we have increased the chance for topic models to find greater use in practical applications.

Besides the novelty of the newly proposed models, we also emphasize the use of concrete methods to evaluate topic models. We verify the performance of the newly proposed topic models against probabilistic Gibbs sampling and variational inference solutions for LDA by applying traditional topic model metrics, such as topic interpretability [Chang et al., 2009, Newman et al., 2010b] and document held-out likelihood [Blei et al., 2003] (Section 2.3). These two metrics are critically important for topic models as indicators of both model quality and topic quality.

1.5.2 Uncovering Insights from Labeled Documents with Supervised Anchor Topic Models

The impressive scalability of the original unsupervised anchor word algorithm brings an interesting research challenge: can we create a *supervised* topic model based on the anchor word algorithm while still keeping this beautiful property of scalability? First, we emphasize that the need for such a model is real. Having supervised topic models that not only run quickly but also produce highly accurate predictions will be very fruitful in helping us understand large labeled datasets. Second, most supervised machine learning models work directly with optimization objectives that minimize the loss function. Solving the optimization objective function gives us a solution that often involves learning associative relationships between independent and dependent variables (class or regression values). These standard supervised learning approaches often do not learn (or do not need to learn) the surface of the input data points. We argue that a combination of supervised models and unsupervised models will create a better machine learning model—a hybrid one. These hybrid models are not only good for making predictions, but are also good for understanding data. In the context of a vast collection of textual content, this combination is a hot research topic under the name of *supervised topic models* [Blei and McAuliffe, 2007, Lacoste-Julien et al., 2009, Ramage et al., 2009, Zhu et al., 2013, Ramage et al., 2011].

Unfortunately, most probabilistic supervised topic models rely on Gibbs sampling or variational inference, which scale poorly to large datasets. Variational inference requires dozens of expensive passes over the entire dataset, and Gibbs sampling requires multiple Markov chains [Nguyen et al., 2014b]. In addition, adding supervised layers to these models slows down their performance because of the additional complicated and expensive inference schemes [Zhu et al., 2013, Wang and Zhu, 2014]. These issues make it extremely hard to put probabilistic supervised topic models to practical use, and practitioners tend to ignore them. We work to create a model as powerful as these probabilistic supervised models but much faster than them. To do so, we improve the anchor word algorithm's extensibility. The extensibility of our topic model allows it to be adapted to more diverse datasets; for example, in this work, extensibility helps the anchor word algorithm to work with labeled datasets. First, we represent the association between label variables and word variables in the form of label vectors; these vectors contain *label-word co-occurrence statistics*. These label vectors are learned using the training set. We then enrich the word vector representation (word embeddings) with these label vectors. The original anchor word algorithm works the same way to learn topics, however, because of the augmented dimensions, topics will capture additional information from metadata, which in turn will help with predicting labels for new documents. We call newly learned topics *supervised topics*, and the anchor word algorithm that works on augmented dimensions the *supervised anchor word* algorithm.[1]

Evaluating on three sentiment datasets, the supervised anchor word algorithm shows superior accuracy performance in comparison to the original anchor word algorithm. The supervised anchor word algorithm also produces more accurate predictions than the probabilistic supervised LDA and unsupervised LDA. Given its fast performance, high topic quality, and superior predictive power, the supervised anchor word algorithm is tremendously helpful for analyzing large labeled datasets. Furthermore, exciting sentiment anchor words learned by the supervised anchor word algorithm provide insights to understand the corpus through sentiment topics: sentiment anchor words explain why and how topics are related to sentiment values, and why documents are classified as such. Lastly, our model can be extended easily for different scenarios such as regression, multi-class classification, and multi-label classification with a simple change of word vector representation. By pro-

[1] There is a clear difference between: (a) using a model to produce the documentation representations, and then training a supervised classifier like SVM for prediction, and (b) using the same method to make the predictions within models like SLDA [Blei and McAuliffe, 2007]. While most probabilistic supervised topic models do (b), our supervised anchor word algorithm does (a).

viding a natural way to embed various types of natural language knowledge and metadata into the process of modeling word representations and enriching their context, our model can adapt to the variability of text data.

1.5.3 The Usability of Supervised Topic Models for Active Learning

Active learning [Settles, 2010, 2012] is a semi-supervised learning mechanism that helps to collect quality labeled information quickly for supervised learning algorithms by presenting a confused data point for users to annotate. The idea is that by collecting those *confused* data points, supervised learning algorithms will be able to learn better with a smaller training set because supervised learning algorithms are directly involved in the labeling step and actively learn the boundaries that distinguish labels. Many active learning strategies have found numerous applications; the two most popular ones are *uncertainty sampling* based on document entropy and *query by committee*. These approaches rely on the core concept of *informativeness* to categorize a confused data point.

In the vein of text applications with large corpora, the ability to quickly perform active learning requires that more insight be gleaned from the corpus of interest. Hence, topic models seem like good candidates for boosting active learning in the realm of classification. Active Learning with Topic Overviews [Poursabzi-Sangdeh et al., 2016, ALTO] is a recently proposed framework for label collection and document labeling that combines uncertainty sampling with topics from LDA. Because ALTO does not change topics as users label more documents, we argue that ALTO misses two crucial points. First, ALTO does not present users with the most updated topics that reflect both labeled and unlabeled documents. Second, ALTO does not incorporate the latest latent topics information into its active learning strategies; these strategies only use the static topics from LDA.

In this new line of work, we develop a new framework based on a fast supervised anchor word algorithm for active learning. This new framework fixes many issues in the ALTO framework. Two big improvements are our incorporation of updated topics and our new sampling strategies that involve document *representativeness* (a concept that emphasizes documents that are representative of the whole corpus). Using updated topic models ensures that active learning strategies have access to the latest information from annotators. Also, adding a representativeness feature into the active learning strategies will prevent the selection of outlier documents, a problem that very often happens in active learning that uses uncertainty sampling [McCallum and Nigam, 1998].[2]

We thoroughly evaluate ALTO and our newly proposed framework using three multi-class datasets and show that our new framework consistently outperforms ALTO in terms of classification accuracy. Additionally, the combination of document informativeness and document representativeness improves the models' performance and robustness.

1.6 Main Technical Contributions

Even though all models introduced in this thesis are motivated by the anchor word algorithm, the key ideas of these models also apply to other inference techniques for topic

[2] Uncertainty sampling represents each document as a probability distribution of labels. This representation resembles topic models in which each document is a probability vector of topics. These two vector representations of documents provide us with more insights and expressiveness to pursue several exciting research directions.

modeling. They can even be generalized for many machine learning models. Additionally, our main applications focus on large-scale textual document collections by introducing several computational methods. We can summarize this thesis with the following technical contributions to statistical analysis and machine learning:

- Introduce and evaluate the concept of priors into the spectral learning methods to improve topic quality, model flexibility, and model robustness (Chapter 3). Our main contribution is to keep the scalability of the original anchor word model while making sure that our new model is on par with probabilistic models in measures of quality and variability. Our work is also the first that systematically evaluates the anchor topic model using standard metrics.

- Extend the fast anchor word algorithm for topic modeling to address the issue of labeled datasets. The newly introduced model captures nuances from the corpus as well as key factors that distinguish label information, hence significantly improving prediction accuracy (Chapter 4). Our contribution extends the richness and flexibility of anchor word models, making them suitable for more applications in the domain of supervised machine learning.

- Combine the concept of supervised topic models and active learning to create a powerful interactive framework. We fix ALTO by amending its active learning strategies to incorporate a *representativeness* concept. Computing representativeness of a document is often fast when using dynamic topic representation from topic models (Chapter 5). Our work contributes significant progress to the domain of interactive machine learning.

Further detailed description of above contributions and their applications will be found in Chapter 6.

Chapter 2

Topic Modeling Foundations

Topic modeling is a subfield of machine learning that extracts the thematic structure from large corpora. Topic models summarize themes that reflect what is being said or written. Topic models are unsupervised algorithms, so by nature, they can explore the structure of data without any guidance (e.g., labels). Given this powerful property, topic models have found numerous applications in many areas such as natural language processing [Boyd-Graber and Blei, 2007, Steyvers and Griffiths, 2007, Wallach, 2008, Mimno et al., 2009], image understanding [Li and Perona, 2005, Luo et al., 2015], dialog systems [Purver et al., 2006, Nguyen et al., 2013b], social media [Yang et al., 2011, Jiang et al., 2015], human-computer interaction [Lee et al., 2017c, Smith et al., 2018], and health care [Wang et al., 2009b, Huang et al., 2014c, Liu et al., 2016].

The most popular topic model is the latent Dirichlet allocation or LDA [Blei et al., 2003]. In LDA, each document is a mixture of multiple topics, and each topic is a distribution over all the words from the vocabulary. LDA is very similar to dimensionality reduction methods [Deerwester et al., 1990, Roweis and Saul, 2000, van der Maaten et al., 2008, Lacoste-Julien et al., 2009, Newman et al., 2010a, Mimno et al., 2011] where we project each document represented as a high dimensional vector (a dimension equal to the number of words) into a much lower dimensional vector (dimension equal to the number of topics). For information retrieval, LDA is much like LSA because the document is represented in low-dimensional space and we can use this representation to compare documents and create document ranking. LDA is also very similar to clustering algorithms such as K-mean clustering; however, in LDA a document can belong to multiple clusters (topics).

LDA is an extension of probabilistic latent semantic analysis [Hofmann, 1999a] for analyzing the statistical properties of co-occurrence data within natural language processing. The fundamental property that distinguishes LDA from previously proposed models is the fact that LDA is the first fully Bayesian model. A fully Bayesian model requires specified prior as compared to an empirical Bayesian model in which we estimate prior based on observed data. LDA uses sparse Dirichlet priors for both document-topic and topic-word priors. Due to Dirichlet priors, a document only contains a small set of topics, and each topic only uses a small set of words more frequently—these properties help LDA better capture written texts.

Since the arrival of LDA, there have been many extensions that enrich the LDA topic model. Examples include adding metadata [Wang and McCallum, 2006, Blei and McAuliffe, 2007, Lacoste-Julien et al., 2009, Ramage et al., 2009, Zhu et al., 2009a, Ramage et al., 2011], creating a topic hierarchy [Blei et al., 2004, Mimno et al., 2007, Blei et al., 2010, Adams et al., 2010, Nguyen et al., 2013c], or combining with other machine learning frameworks [Hu et al., 2014a, Hinton and Salakhutdinov, 2009, Larochelle and Lauly, 2012, Poursabzi-Sangdeh et al., 2016]. Current big datasets also introduce many more challenging problems for topic modeling research. Many new solutions address issues such as scalability [Nallapati et al., 2007, Smola and Narayanamurthy, 2010, Zhai et al., 2012, Wang et al., 2011, Wang and Blei, 2012, Bryant and Sudderth, 2012, Li et al., 2014, Hoffman et al., 2013], multi-modality [Putthividhy et al., 2010, Virtanen et al., 2012, Nguyen et al., 2013a, Qian et al., 2016], and large-scale visualization [Gardner et al., 2010, Gretarsson et al., 2012, Chaney and Blei, 2012, Chuang et al., 2012].

This chapter provides relevant background on the active research area of topic models. We first introduce the basic concepts of topic modeling in Section 2.1. We then describe LDA in-depth, first introducing its generative process, and then deriving typical inference schemes for training LDA in Section 2.1.2. Some popular extensions of LDA to address issues such as scalability of topic modeling, supervision, incorporating external knowledge, or interaction with users will also be introduced in Section 2.4. Furthermore, we describe two popular metrics for evaluating topic models: the document held-out likelihood and the topic interpretability. While the held-out likelihood of a model measures how well the model generalizes, topic quality measurement by topic interpretability associates with how a human understands topics produced by topic models.

The second part of this background chapter describes a spectral method called the *anchor word algorithm* for topic models [Arora et al., 2012b, 2013]. The anchor word algorithm is a novel inference scheme for LDA and serves as the foundational background for this thesis. The key idea of this algorithm is the application of the non-negative matrix factorization (NMF) with a separability assumption [Donoho and Stodden, 2004] on the topic matrix. The anchor word algorithm is also the first topic model algorithm that provides (theoretically) a provable guarantee: it ensures finding the global topic matrix given a sufficient number of documents.

To understand the anchor algorithm, we will first start with an introduction to non-negative matrix factorization and some of its useful properties for topic modeling in

Section 2.5.1. Following the background on NMF, we describe detailed steps of the anchor word algorithm in Section 2.6. We then point out some advantages and disadvantages of the anchor algorithm in comparison to the current probabilistic inference schemes such as the Gibbs sampling and the mean-field variational method. Finally, we describe our contributions in this thesis through extending the anchor algorithm to address scalability, variability, and interactivity when applying topic models for real-world applications.

2.1 What is Topic Modeling?

In this section, we describe some general concepts of topic models. Because this thesis emphasizes textual documents, we tailor our description of these concepts toward text domain.

2.1.1 Topic Definition

A *topic* is visually represented as a coherent set of words. This set of words conveys the meaning of a topic, and we understand what a topic is about just by seeing these words. Table 2.1 shows two example topics. A topic about "soccer" contains words such as "goal", "soccer", or "net" because these words often appear together in documents, and they semantically construct a solid meaning about the sport of soccer. As we see in the later sections, topic models learn these concepts for topics using deeper (second- or even third-order) associations. For example, "fifa" and "net" do not often appear together in many documents; however, because they are both used in the context of "soccer", they go along with each other in a topic about soccer.

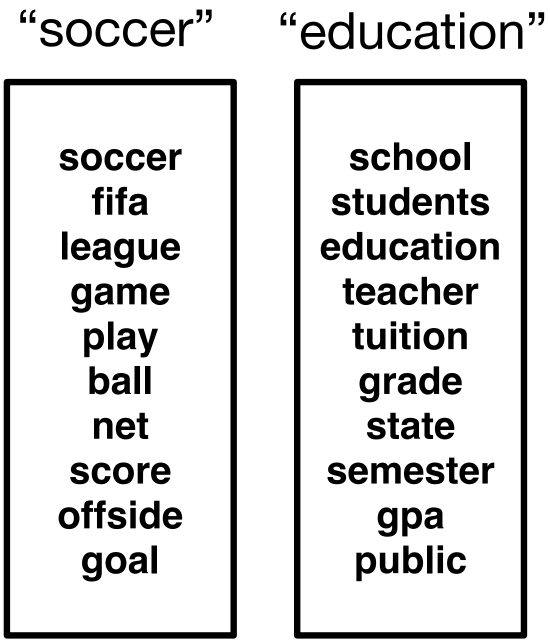


Figure 2.1: Examples of how a topic looks like. On the left, the soccer topic contains related words about soccer. On the right, the education topic contains related words about school system.

Because most topic models are motivated by the original, probabilistic latent Dirichlet allocation model, many terminologies come from the LDA model. In LDA, a topic is a distribution of all the words in the vocabulary, and we often use the most probable words in a topic distribution to visually represent a topic. For example, the list of words in the Figure 2.1 contains the top ten words that have the highest probabilities in a topic. The typical representation of topics in many topic models is pretty consistent and under the name of *topic-word matrix* each column corresponds to one topic distribution over all the words. Hence, a topic-word matrix is also called *topic-word distributions*. The topic-word matrix is the required output of every topic modeling algorithm.

In addition to the topic-word matrix, topic modeling algorithms also produce other types of output—depending on the context and the application domain. For example, like LDA, most topic models assume a document is a mixture of topics and return one document-topic distribution for each document. For all documents from a corpus, the output is a matrix and is called *document-topic matrix* or *document-topic distributions*. However, this is not the only view for the relationships of documents and topics. Some initial topic models such as the probabilistic latent semantic analysis [Hofmann, 1999a] assume that a document only contains one topic.[1] Also, other types of topic models may return specific outputs such as labels or real values for prediction purposes (in supervised topic models), document relationships (in relational topic models), or topic covariance (in the pachinko allocation model [Mimno et al., 2007]). These types of outputs are important to show the wide applications of topic models as well as their flexibility to capture other types of information or to perform diverse tasks. We will describe some of these rich topic models later in this chapter as they motivate our solutions in this thesis.

Let us go into detail of the LDA model introduced by Blei et al. [2003].

2.1.2 Latent Dirichlet Allocation

This section describes the most popular topic model: Latent Dirichlet Allocation or LDA [Blei et al., 2003]. LDA is a probabilistic mixture model where each document is a mixture of topics, and each topic is a distribution over all the words in the vocabulary. The mixture of topics per document is the crucial difference that makes LDA becomes popular. Before LDA, models such as the latent semantic analysis (LSA) or the probabilistic latent semantic analysis (PLSA) assume only one topic per document [Papadimitriou et al., 1998, Hofmann, 1999a]. In addition, LDA is a fully Bayesian model where we specify the

[1] If we ignore the topic matrix, clustering algorithms can be considered a special topic model.

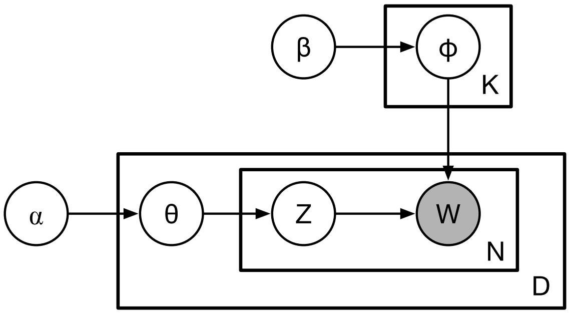


Figure 2.2: Generative process of LDA using "plates" representation. The boxes represent replicates. At the top, the box represents topics (K of them). At the bottom, the outer plate represents documents (D of them), while the inner plate represents choices of topics and words within a document [Blei et al., 2003].

model's prior; it is a more powerful method than PLSA. The beauty of LDA also lies in its well-defined generative process and its generalization to new documents (which is because of the fully Bayesian characteristic).

LDA assumes a Dirichlet distribution as prior for a document over topics and a topic over words. These Dirichlet priors are critical for LDA. Using Dirichlet priors, each document is only represented by a small number of topics, and at the same time, each topic uses only a small number of words. Because of this, LDA captures latent topics and composition of documents better than other models. As we will also see later, this Dirichlet prior assumption is crucial for speeding up the inference of LDA because of the conjugacy between a Dirichlet distribution and a Multinomial distribution [Blei et al., 2003].

We first describe the generative process of LDA in which documents are generated through a stochastic process using Dirichlet priors.

LDA Generative Process To understand the generative process of LDA, we first introduce some notation. We will use these notations throughout the thesis. We use D as the number of documents in the corpus of interest. The corpus vocabulary size is V. And N_d is the number of words in a document d. LDA also assumes the number of topics to be fixed and is equal to K. The generative process (Figure 2.2) is as the following:

1. Draw K topic over words vectors: $\beta_k \sim \text{Dir}_V(\boldsymbol{\eta})$
2. For each document $d = 1, \ldots, D$ do
    - Draw a document length: $N_d = \text{Poison}(N)$
    - Draw a topic proportions: $\theta_d \sim \text{Dir}_K(\boldsymbol{\alpha})$
    - For each word $n = 1, \ldots, N_d$ do
        - Draw a topic assignment: $z_{d,n} = \text{Multinomial}(\theta_d)$, $z_{d,n} \in [K]$.
        - Draw a word: $w_{d,n} = \text{Multinomial}(\beta_{z_{d,n}})$, $w_{d,n} \in [V]$.

Here vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ are the Dirichlet hyper-parameters that control the priors on document topic and topic word distributions. N is the hyper-parameter for the Poisson prior of document length.

Dirichlet Distribution A Dirichlet distribution [Minka, 2000b] is a distribution over a set of finite discrete probability distribution. Given a parameter $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ of non-negative values, the Dirichlet distribution, denoted by $\text{Dir}(\boldsymbol{\alpha})$, will generate a sample as a probability distribution. Each sample from the Dirichlet distribution is a vector of weights whose sum is equal to one. The sample distribution drawn from a Dirichlet distribution is often used as a parameter for a multinomial distribution. The notation of a multinomial distribution in the above generative process is equivalent to the categorical distribution [Murphy, 2012] where each draw is an index. The density function of a Dirichlet distribution is

$$
\begin{aligned}
p(x \mid \boldsymbol{\alpha}) &= \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} x_k^{\alpha_k - 1} \\
&= \frac{1}{\Delta(\boldsymbol{\alpha})} \prod_{k=1}^{K} x_k^{\alpha_k - 1},
\end{aligned}
\tag{2.1}
$$

where $\Gamma(x)$ is the Gamma function, and $\Delta(\boldsymbol{\alpha})$ is the Dirichlet Delta function (sometimes it is called *multivariate Beta function*). Because $\int p(x \mid \boldsymbol{\alpha}) dx = 1$, from Equation 2.1, we have

$$
\Delta(\boldsymbol{\alpha}) = \int \prod_{k=1}^{K} x_k^{\alpha_k - 1} dx. \tag{2.2}
$$

The Gamma function is an extension of the factorial function, when x is a positive integer, $\Gamma(x) = (x-1)!$. When x is a real or complex with positive real part, then

$$
\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt.
$$

When all the values of α_is are equal to α_0, the Dirichlet distribution is *symmetric* and is often denoted by Dir$(\boldsymbol{\alpha_0})$. In other cases, we call Dirichlet distribution an *asymmetric* distribution.[2]

Going back to the generative process of LDA, for a given document d, the document-topic distribution θ_d drawn from the Dirichlet$(\boldsymbol{\alpha})$ will be the parameter for a multinomial

[2] Sample draws from a Dirichlet distribution are mostly sparse if these α parameters are less than one [Telgarsky, 2013].

distribution: Multinomial(θ_d). The topic index for a word $w_{d,n}$ follows the density

$$p(z_{d,n} \,|\, \theta_d) = \prod_{k=1}^{K} \theta_{d,k}^{[z_{d,n}=k]}. \tag{2.3}$$

The Dirichlet distribution belongs to the exponential family of distributions and is conjugate to the multinomial distribution. Conjugacy means that the posterior also has the Dirichlet distribution, and its formulation is derived directly from the parameters of the prior function and the likelihood function [Minka, 2000a]. A conjugate property facilitates the development of inference algorithms for LDA. Before going into the in-depth discussion of how inference algorithms for LDA work, we will briefly describe the mathematical formula for the likelihood function of LDA. When they all come together, the likelihood function and Equation 2.3 will help us to derive the inference solutions for LDA.

Likelihood We formulate the likelihood of generating the documents assuming that documents are generated by the LDA's generative process as described in Figure 2.2.

Given a hyper-parameter vector $\boldsymbol{\eta}$ of size V, each vector β_k is a distribution over all the V words in the vocabulary and its density similarly follows the Dirichlet distribution.

$$p(\beta_k \,|\, \boldsymbol{\eta}) = \frac{\Gamma(\sum_{v=1}^{V} \eta_v)}{\prod_{v=1}^{V} \Gamma(\eta_v)} \prod_{v=1}^{V} \beta_{k,v}^{\eta_v - 1}. \tag{2.4}$$

The generation of β is independent of how documents are generated; therefore we can safely assume β is fixed before the document generation process. Notation-wise, β is a matrix of size K by V.

Similarly, given a hyper-parameter $\boldsymbol{\alpha}$ of size K, each vector θ_d for a document d is a distribution over K topics. θ_d is drawn from $\mathrm{Dir}(\boldsymbol{\alpha})$ so

$$p(\theta_d \mid \boldsymbol{\alpha}) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{d,k}^{\alpha_k - 1}. \tag{2.5}$$

Given these distributions from Equation 2.4 and Equation 2.5, the joint probability for a document d of length N_d of words w_d and their topic assignments z_d is

$$p(w_d, z_d, \theta_d \mid \beta; \boldsymbol{\alpha}, \boldsymbol{\eta}) = p(\theta_d \mid \boldsymbol{\alpha}) \prod_{n=1}^{N_d} p(z_{d,n} \mid \theta_d) p(w_{d,n} \mid \beta_{z_{d,n}}). \tag{2.6}$$

For all D documents from a corpus, we have the likelihood of all of them is

$$p(w, z, \theta \mid \beta; \boldsymbol{\alpha}, \boldsymbol{\eta}) = \prod_{d=1}^{D} p(\theta_d \mid \boldsymbol{\alpha}) \prod_{n=1}^{N_d} p(z_{d,n} \mid \theta_d) p(w_{d,n} \mid \beta_{z_{d,n}}). \tag{2.7}$$

We can integrate out all the θs, and sum over all topic assignments $z_{d,n}$ to get a simple formulation for the likelihood as

$$p(w \mid \beta; \boldsymbol{\alpha}, \boldsymbol{\eta}) = \prod_{d=1}^{D} \int p(\theta_d \mid \boldsymbol{\alpha}) \prod_{n=1}^{N_d} p(w_{d,n} \mid \theta_d, \beta) \nabla \theta_d, \tag{2.8}$$

where $p(w_{d,n} \mid \theta_d, \beta) = \sum_{z_{d,n}} p(z_{d,n} \mid \theta_d) p(w_{d,n} \mid \beta_{z_{d,n}})$.

Posterior Inference LDA belongs to a class of latent variable models; to learn the latent variables the usual method is to estimate the posterior distribution based on the prior and the likelihood functions. Specifically, in the inference step of LDA, our goal is to estimate the βs, θs, and topic assignment z based on the observed documents and hyper-parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$. The global topics β_k captures what are talked about in the entire corpus. The document-specific topic proportions θ_d captures salient topics that are talked about locally in each document. Since the inference is intractable, we often have to use approximate methods such as Gibbs sampling and mean-field variational inference. In the next section, we will derive in detail each of these popular inference schemes. Learning hyper-parameters or parameter estimation is also critical for LDA applications and we also briefly touch on that issue in the later sections.

Notation Custom Before delving into the detail of inference derivation, we use the following customs from both Blei et al. [2003] and Heinrich [2004]. Basically

- For a word, $w_{d,n}$ is a one-hot vector of size V where only one component is one, and all other components are zero. For example, when we write $w_{d,n} = v$, we mean that the v component, which is denoted by $w_{d,n}^v$, has value of one or $w_{d,n}^v = 1$.
- Even though each document has a different number of words, and its length is N_d, we will use N (the maximum document length of all documents in a corpus) as the only length for any document. This custom is used in most LDA related papers. Implementation-wise if a document d does not contain a word n, then there is no $w_{d,n}$ anywhere in equations.

2.2 Inference Methods for Topic Models

In this section, we describe the two most popular inference methods for learning topics from the latent Dirichlet allocation: Gibbs sampling and variational inference. By digging deeper into the mechanics of each method, we point out their strength and weaknesses. While Gibbs sampling approaches provide us with a good solution for LDA and guarantee converging to correct model distributions, they often take a long time to do so. Gibbs sampling fixes often sacrifice this guaranteed correctness for performance and scalability. On the other hand, variational approaches provide us with more flexibility and scalability to consume and digest huge document collections. However, variational solutions are approximate, and there is no guarantee of convergence to a correct model distribution. Depending on actual text applications these approaches will find their usefulness.

2.2.1 Gibbs Sampling

Gibbs sampling is a particular method within the realm of the Markov chain Monte Carlo (MCMC) simulations. Gibbs sampling has proved itself as a simple, yet effective solution for LDA. The main power of Gibbs sampling comes from the general MCMC techniques that are often applied to solve integration and optimization problems in high-dimensional spaces. These problems are prevalent in machine learning due to the large size of data. MCMC techniques assume that we know the distribution and want to draw samples from it [Heinrich, 2004, Neal, 1993, 2000, Robert and Casella, 2004, Johnson et al., 2007, Griffiths and Steyvers, 2004, McCallum, 2002, Resnik and Hardisty, 2010].

MCMC The main idea of MCMC techniques is to draw many *i.i.d* sets of samples from a target density of posterior distribution using the stationary behavior of a Markov chain. In a Markov chain, there are many states (finite, countably finite, or infinite) and walking across states is called a *transition*. A state can be anything, for example, an assignment of a specific topic for a word in a document. A state space is a list of states.

MCMC performs many simulations to walk many steps to visit states. As we walk many, many times, we will start passing a state called *the stationary state*, where every transition in a chain after that state will correspond to one sample from the desired distribution. By exploring the state space using a Markov chain, MCMC spends more time in the most important regions of the distribution. The chain will eventually converge to a stable invariant distribution if the following conditions are satisfied:

- *Irreducibility* The transition matrix is connected.
- *Aperiodicity* The chain should not get trapped in cycles.

Gibbs Sampling Procedure Gibbs sampling requires that we have all the full conditional distributions of the target posterior distribution to be sampled. Let us say that we would like to sample a n dimensional vector variable $\boldsymbol{x}$ that follows the distribution $p(\boldsymbol{x})$, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) = x_{1:n}$. Denote x_{-i} being $(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, then $p(x_i \,|\, x_{-i})$ is called the full conditional probability. Given these conditional probabilities, we will sample each dimension x_i alternately conditioned on other dimensions. We follow Andrieu et al. [2003] to describe a general procedure of Gibbs sampling to sample for variables $x_{1:n}$ as the following:

1. Initialize $x^0_{1:n}$.

2. For $i = 0$ to $L - 1$ do

   - Sample $x^{i+1}_1 \sim p(x_1 \,|\, x^i_2, x^i_3, \ldots, x^i_n)$.
   - Sample $x^{i+1}_2 \sim p(x_2 \,|\, x^{i+1}_1, x^i_3, \ldots, x^i_n)$.
   - $\ldots$
   - Sample $x^{i+1}_j \sim p(x_j \,|\, x^{i+1}_1, \ldots, x^{i+1}_{j-1}, x^i_{j+1}, \ldots, x^i_n)$.
   - $\ldots$
   - Sample $x^{i+1}_n \sim p(x_n \,|\, x^{i+1}_1, x^{i+1}_2, \ldots, x^{i+1}_{n-1})$.

So the general premise of Gibbs sampling is the usage of the full conditional probability to draw samples. After having samples, we take an average of them to get desire estimation for variables.

Implementing a Gibbs sampling procedure requires an understanding about other issues as well. For example, Resnik and Hardisty [2010] give a detailed list of things to look for when implementing a Gibbs sampling procedure; two factors to consider are *burn-in* and *lag*. *Burn-in* means to avoid early samples before we reach the stationary state of the chain; after burn-in, samples are correctly coming from the stationary distribution. *Lag* is an important concept to avoid collecting autocorrelated samples in successive steps; we take an average only every *lag* steps [Resnik and Hardisty, 2010, Nguyen et al., 2014b].

In the next section, we will use the above Gibbs sampling procedure to sample topic assignment ($\boldsymbol{z}$) and other variables of the LDA model. We follow guidance on how to do Gibbs sampling for general latent variable models from Resnik and Hardisty [2010] and use some detailed Gibbs sampling derivation of LDA in Heinrich [2004]. We only describe the most essential derivation; more recent methods focus on creating fast Gibbs sampling procedures for LDA [Porteous et al., 2008, Zhu et al., 2013] and are beyond the scope of this chapter.

Gibbs Sampling for LDA Variables that we need to estimate for an LDA model are the topic assignment, the topic-word matrix, and the document-topic matrix.

The Topic Assignment of Words

For the topic assignment of each word in a document, we would like to estimate $p(z_{d,n} = k)$ for each document d, each word n in d, and each topic k.

Denote $\boldsymbol{w}$ as all words in all documents in the corpus and $\boldsymbol{z}$ as the topic assignment of those words. Set $I = |w|$, is the count of all words in all documents in the corpus. The posterior distribution of topic assignment $\boldsymbol{z}$ given all words $\boldsymbol{w}$ is

$$p(\boldsymbol{z} \mid \boldsymbol{w}) = \frac{p(\boldsymbol{z}, \boldsymbol{w})}{p(\boldsymbol{w})} = \frac{\prod_{i=1}^{I} p(z_i, w_i)}{\prod_{i=1}^{I} \sum_{k=1}^{K} p(z_i = k, w_i)},$$

here each topic assignment for a word w_i is z_i, $z_i \in [1, K]$.

Due to the intractable computation of the denominator of the above formula, we use Gibbs sampling based on the full conditional $p(z_i \mid z_{-i}, w)$. We derive formulae of these conditional probabilities based on the formula for the joint distribution $p(\boldsymbol{z}, \boldsymbol{w})$, by Bayes'rule

$$p(\boldsymbol{z}, \boldsymbol{w} \mid \boldsymbol{\alpha}, \boldsymbol{\eta}) = p(\boldsymbol{w} \mid \boldsymbol{z}, \boldsymbol{\eta}) p(\boldsymbol{z} \mid \boldsymbol{\alpha}).$$

Because the distribution of words given a topic is multinomial, we have

$$p(\boldsymbol{w} \mid \boldsymbol{z}, \beta) = \prod_{k=1}^{K} \prod_{v=1}^{V} \beta_{k,v}^{n_{k,v}}, \tag{2.9}$$

where $n_{k,v}$ is the number of times word v has been assigned to topic k. In order to estimate the distribution $p(\boldsymbol{w} \mid \boldsymbol{z}, \boldsymbol{\eta})$, we integrate out latent parameters β to get a closed-form formula. Using the above Equation 2.9 we have

$$\begin{aligned} p(\boldsymbol{w} \mid \boldsymbol{z}, \boldsymbol{\eta}) &= \int p(\boldsymbol{w} \mid \boldsymbol{z}, \beta) p(\beta \mid \boldsymbol{\eta}) d\beta \\ &= \int \prod_{k=1}^{K} \frac{1}{\Delta(\boldsymbol{\eta})} \prod_{v=1}^{V} \beta_{k,v}^{n_{k,v}+\eta_v-1} d\beta_k \\ &= \prod_{k=1}^{K} \frac{1}{\Delta(\boldsymbol{\eta})} \int \prod_{v=1}^{V} \beta_{k,v}^{n_{k,v}+\eta_v-1} d\beta_k \\ &= \prod_{k=1}^{K} \frac{\Delta(\boldsymbol{n}_k + \boldsymbol{\eta})}{\Delta(\boldsymbol{\eta})}, \end{aligned} \tag{2.10}$$

where $\boldsymbol{n}_k = \{n_{k,v}\}_{v=1}^{V} = (n_{k,1}, \ldots, n_{k,V})$. Both $\boldsymbol{n}_k$ and $\boldsymbol{\eta}$ are V dimensional vectors. The above cancelation is due to Equation 2.2.

Similarly, the conditional probability of topic assignment $\boldsymbol{z}$ given $\boldsymbol{\alpha}$ is computed by integrating over all θ. First, the multinomial distribution of topic assignment given θ is

$$p(\boldsymbol{z} \mid \theta) = \prod_{i=1}^{I} p(z_i \mid d_i) = \prod_{d=1}^{D} \prod_{k=1}^{K} p(z_i = k \mid d_i = d) = \prod_{d=1}^{D} \prod_{k=1}^{K} \theta_{d,k}^{m_{d,k}},$$

where d_i denotes the document that the word w_i belongs to and $m_{d,k}$ are the number of times that topic k appears with a word in document d. Integrating out θ, we have

$$\begin{aligned} p(\boldsymbol{z} \mid \boldsymbol{\alpha}) &= \int p(\boldsymbol{z} \mid \theta) p(\theta \mid \boldsymbol{\alpha}) d\theta \\ &= \int \prod_{d=1}^{D} \frac{1}{\Delta(\boldsymbol{\alpha})} \prod_{k=1}^{K} \theta_{d,k}^{m_{d,k}+\alpha_k-1} d\theta_d \\ &= \prod_{d=1}^{D} \frac{\Delta(\boldsymbol{m}_d + \boldsymbol{\alpha})}{\Delta(\boldsymbol{\alpha})}, \end{aligned} \tag{2.11}$$

where $\boldsymbol{m}_d = \{m_{d,k}\}_{k=1}^{K} = (m_{d,1}, \ldots, m_{d,K})$. Both $\boldsymbol{m}_d$ and $\boldsymbol{\alpha}$ are K dimensional vectors.

From Equation 2.10 and Equation 2.11, the joint distribution of topic assignment z and words w is

$$p(\boldsymbol{z}, \boldsymbol{w} \mid \boldsymbol{\alpha}, \boldsymbol{\eta}) = \prod_{k=1}^{K} \frac{\Delta(\boldsymbol{n}_k + \boldsymbol{\eta})}{\Delta(\boldsymbol{\eta})} \times \prod_{d=1}^{D} \frac{\Delta(\boldsymbol{m}_d + \boldsymbol{\alpha})}{\Delta(\boldsymbol{\alpha})} \tag{2.12}$$

Because Gibbs sampling relies on full conditional probabilities, we need to compute the full conditional distribution for each word and its topic assignment (z_i, w_i), given that all other elements are fixed.

Follow Heinrich [2004], the full conditional distribution for a topic assignment z_i given word observation $w_i = v$ and all others fixed is

$$\begin{aligned} p(z_i = k \mid \boldsymbol{z}_{-i}, \boldsymbol{w}) &= p(z_i = k \mid \boldsymbol{z}_{-i}, w_i = v, \boldsymbol{w}_{-i}) \\ &= \frac{p(\boldsymbol{w}, \boldsymbol{z})}{p(\boldsymbol{w}, \boldsymbol{z}_{-i})} \\ &= \frac{p(\boldsymbol{w} \mid \boldsymbol{z})}{p(\boldsymbol{w}_{-i} \mid \boldsymbol{z}_{-i}) p(w_i)} \times \frac{p(\boldsymbol{z})}{p(\boldsymbol{z}_{-i})} \\ &\propto \frac{n_{k,v,-i} + \eta_v}{\sum_{v=1}^{V} n_{k,v,-i} + \eta_v} \times \frac{m_{d,k,-i} + \alpha_k}{[\sum_{k=1}^{K} m_{d,k} + \alpha_k] - 1}, \end{aligned} \tag{2.13}$$

where $n_{k,v,-i}$ denotes that the token at position i is excluded from topic when we do the counting. Also $m_{d,k,-i}$ denotes that we do not count topic assignment for the token at position i in document d. The formula means that the probability of assigning a topic k for a word i in document d is proportional to how likely others words have been assigned to topic k globally across all documents and how many times topic k has been observed with other words (rather than i) within the document d (locally).

The Topic Word Matrix

Next, we will compute the multinomial parameters of the topic word matrix β. It is rather straightforward, given the n counts we have for β:

$$p(\boldsymbol{\beta}_k \,|\, \boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\eta}) \sim \mathrm{Dir}(\boldsymbol{\beta}_k \,|\, \boldsymbol{n}_k + \boldsymbol{\eta}).$$

The neat form above comes directly from the nice conjugate property of the Dirichlet distribution with the multinomial distribution.

By the expectation of the Dirichlet distribution, $\mathbb{E}\left[\mathrm{Dir}(x)\right]_i = x_i / \sum_j x_j$, we have

$$\beta_{k,v} = \frac{n_{k,v} + \eta_v}{\sum_{v=1}^{V} n_{k,v} + \eta_v}. \tag{2.14}$$

The Document Topic Matrix

Similarly for θ

$$p(\boldsymbol{\theta}_d \,|\, \boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\alpha}) \sim \mathrm{Dir}(\boldsymbol{\theta}_d \,|\, \boldsymbol{m}_d + \boldsymbol{\alpha}).$$

And by expectation,

$$\theta_{d,k} = \frac{m_{d,k} + \alpha_k}{\sum_{k=1}^{K} m_{d,k} + \alpha_k}. \tag{2.15}$$

Using Equation 2.13, Equation 2.14, and Equation 2.15, we can now run a Gibbs sampling algorithm to find the solution for the LDA model.

For implementation, there are three count matrices that we need to use to keep track of samples. The first matrix is the topic word counts $n_{k,v}$ of dimension $K \times V$. The second matrix is the document topic counts $m_{d,k}$ of dimension $D \times K$. The final matrix is the topic assignment variable $z_{d,n}$. The total counts of z is W; however, we often use a matrix of dimension $D \times N$ where N is the maximum length of all documents in the corpus (also described in the generative process of LDA).

2.2.2 Variational Inference

In this section, we describe a mean-field variational method for approximate posterior inference of LDA. As introduced by Jordan et al. [1999], variational methods provide approximate inference algorithms for graphical models. The general idea is to replace the posterior distribution with a proxy distribution. A proxy distribution is called a variational distribution. In variational inference (VI), using a variational distribution removes the dependencies among hidden variables, making the new inference problem more tractable. The main reason is that the variational distribution decomposes into multiple separate factors. Variational methods are also closely related to the bound from the convexity. For example, a complex concave function can always be upper-bounded by a linear function of additional parameters. In variational inference or convex relaxation, the focus question is to learn the best surrogate parameters of the replaced distribution or function. Solutions often involve solving a corresponding optimization problem.

Variational Method We follow Blei et al. [2016] to describe the general concept and solution of the variational method for approximating any posterior inference.

Given the observed data x, the problem of posterior inference is to estimate parameters of the posterior distribution $p(z \mid x)$. The goal of variational method is to find a variational distribution $q(z)$ that can approximate the posterior distribution $p(z \mid x)$, such as the computation becomes easier with q than with p. The distribution q often belongs to a family of densities $\mathcal{Q}$. In VI, Kullback-Leibler divergence (or in short, KL divergence) is used to measure the distance between two distributions. Using KL divergence, our goal is to find the best variational distribution $q^*(z)$:

$$\begin{aligned} q^*(z) &= \text{argmin}\left(D_{\text{KL}}\left(q(z) \,||\, p(z \mid x)\right)\right) \\ &= \text{argmin}\left(\mathbb{E}_q\left[\log(q(z))\right] - \mathbb{E}_q\left[\log(p(z \mid x))\right]\right) \\ &= \text{argmin}\left(\mathbb{E}_q\left[\log(q(z))\right] - \mathbb{E}_q\left[\log(p(z, x))\right] + \log(p(x))\right), \end{aligned} \tag{2.16}$$

where $\mathbb{E}_q\left[*\right]$ denotes the expectation with respect to $q(z)$ distribution.

In the above Equation 2.16, finding the best density by minimizing the KL divergence is equivalent to maximize the evidence of lower bound (ELBO) function: $\text{ELBO}(q) = \mathbb{E}_q\left[\log(p(z, x))\right] - \mathbb{E}_q\left[\log(q(z))\right]$. The ELBO function is an important concept in variational inference. ELBO is the main function that we will try to maximize to find the best variational density.

Let us first try to see what it really means to maximize ELBO(q) by rewriting it as

$$\begin{aligned}\text{ELBO}(q) &= \mathbb{E}_q\left[\log(p(z,x))\right] - \mathbb{E}_q\left[\log(q(z))\right] \\ &= \mathbb{E}_q\left[\log(p(z))\right] + \mathbb{E}_q\left[\log(p(x \mid z))\right] - \mathbb{E}_q\left[\log(q(z))\right] \\ &= \mathbb{E}_q\left[\log(p(x \mid z))\right] - D_{\textbf{KL}}\left(q(z) \,||\, p(z)\right).\end{aligned} \tag{2.17}$$

In Equation 2.17, the first term $\mathbb{E}_q\left[\log(p(x \mid z))\right]$ is the expectation log of the likelihood; so maximizing this term means that we encourage densities that place their mass on latent variables that maximally explain observed data. Also, minimizing the second term $D_{\textbf{KL}}\left(q(z) \,||\, p(z)\right)$ means that we encourage the variational distribution to be close to the prior distribution as much as possible. So maximizing the ELBO function balances between explaining the observed data and following the trajectory of the prior distribution.

Mean-Field Family

One crucial aspect of variational inference is the choice of the density family $\mathcal{Q}$. This family determines how complex it is to use ELBO as an objective function for finding approximate solutions. Most current solutions in topic modeling focus on the mean-field variational inference family where latent variables are mutually independent and each is governed by a distinct factor. In this family, the variational density decomposes into factors:

$$q(z) = \prod_{j=1}^{m} q_j(z_j), \tag{2.18}$$

where each variational factor $q_j(z_j)$ is independent of each other. Using independent factors simplifies the solution and reduces the complexity of the optimization problem of maximizing the ELBO(q).[3]

Specifically,

$$\begin{aligned}
q_j^*(z_j) &= \text{argmax}\left(\text{ELBO}(q)\right) \\
&= \text{argmax}\left(\mathbb{E}_q\left[\log(p(z,x))\right] - \mathbb{E}_q\left[\log(q(z))\right]\right) \\
&= \text{argmax}\left(\mathbb{E}_q\left[\log(p(z_j, z_{-j}, x))\right] - \sum_{j=1}^{m} \mathbb{E}_q\left[\log(q_j(z_j))\right]\right) \\
&= \text{argmax}\left(\mathbb{E}_j\left[\mathbb{E}_{-j}\left[\log(p(z_j, z_{-j}, x))\right]\right] - \mathbb{E}_j\left[\log(q_j(z_j))\right]\right) \\
&\propto \exp\left(\mathbb{E}_{-j}\left[\log(p(z_j, z_{-j}, x))\right]\right) \\
&\propto \exp\left(\mathbb{E}_{-j}\left[\log(p(z_j \mid z_{-j}, x))\right]\right),
\end{aligned} \tag{2.19}$$

where $\mathbb{E}_{-j}\left[*\right]$ means that the expectation is computed over all $q_i(z_i)$s and $i \neq j$.

CAVI Algorithm

The general procedure for finding q using a mean-field variational inference is very similar to what we have described for the Gibbs sampling. We iteratively optimize each factor of q while holding other factors fixed. Algorithm 1, using Equation 2.19, describes the famous *coordinate ascent variational inference* (CAVI) algorithm [Blei et al., 2016].

Solution of CAVI (coordinate ascent variational inference) provides variational parameters for each variational factor $q_j(z_j)$. Given variational parameters, we would use the found (best) variational distribution as if we would use the posterior distribution.

CAVI has two important terms that we need to compute: the expected log of

[3] Using mean-field family to approximate intractable posterior in variational inference is similar to using a convex function to approximate non-convex function in convex relaxation.

Algorithm 1 CAVI algorithm

Input: A model $p(x, z)$, a data x
Output: A variational density $q(z) = \prod_{j=1}^{m} q_j(z_j)$

```
while the ELBO has not converged do
    for $j \in \{1, \ldots, m\}$ do
        Set $q_j(z_j) \propto \exp\left(\mathbb{E}_{-j}\left[\log(p(z_j \mid z_{-j}, x))\right]\right)$
    Compute $\mathrm{ELBO}(q) = \mathbb{E}_q\left[\log(p(z, x))\right] - \mathbb{E}_q\left[\log(q(z))\right]$
return $q(z)$
```

the complete conditional function $\mathbb{E}_{-j}\left[\log(p(z_j \mid z_{-j}, x))\right]$ and ELBO. It is not always straightforward to compute these two terms due to the complexity of the chosen density family and the likelihood functions. In practice, in order to use CAVI, we often settle on a particular family of distributions so that computing these two terms are tractable and fast. One of them is the exponential family of distributions, which we will use for the derivation of the LDA model.

Variational Inference for LDA Computing posterior approximations for LDA relies on the closed-form formula of variational parameters of variational densities that belong to a unique family of distributions: the *exponential family*. As described in the generative process of LDA, LDA uses two distributions: the Dirichlet distribution and the multinomial distribution. Because both distributions belong to the exponential family, we can compute their variational parameters directly.

We list variational parameters for the three posterior distributions of interest: the topic assignment $z_{d,n}$, the document-topic matrix $\theta_{d,k}$, and the topic-word matrix $\beta_{k,v}$. More in-depth step-by-step derivations for the whole LDA model can be found in the Appendix of Blei et al. [2003].

Algorithm 2 CAVI algorithm for LDA

```
Input: LDA model and a set of words in documents $w$
Output: Variational parameters $\lambda, \gamma, \phi$
Initialize: Variational parameters $\lambda, \gamma$ randomly

while the ELBO has not converged do
    while $\phi$ and $\gamma$ has not converged do
        for each document d do do
            for each word n do do
                Update $\phi^k_{d,n} \propto \exp\Big(\Psi(\gamma_{d,k}) + \Psi(\lambda_{k,w_{d,n}}) - \Psi(\sum_{v=1}^{V} \lambda_{k,v})\Big)$
                Update $\gamma_d = \alpha + \sum_{n=1}^{N} \phi_{d,n}$
    Update $\lambda_k = \eta + \sum_{d=1}^{D} \sum_{n=1}^{N} \phi^k_{d,n} w_{d,n}$
    Update ELBO
```

Locally, for each document d, we iterate over all the words in d and update the variational parameters for the topic assignment and the document topic matrix:

$$\phi^k_{d,n} \propto \exp\left(\Psi(\gamma_{d,k}) + \Psi(\lambda_{k,w_{d,n}}) - \Psi(\sum_{v=1}^{V} \lambda_{k,v})\right). \tag{2.20}$$

$$\gamma_d = \alpha + \sum_{n=1}^{N} \phi_{d,n}. \tag{2.21}$$

The above updates above depend on the variational parameters of the variational parameters for topic word matrix λ:

$$\lambda_k = \eta + \sum_{d=1}^{D} \sum_{n=1}^{N} \phi^k_{d,n} w_{d,n}. \tag{2.22}$$

We update the exact CAVI algorithm for LDA as in Algorithm 2.

Which Methods Should We Use: VI or MCMC? One of the immediate advantages of the variational inference method in comparison to Gibbs sampling is the rich modeling capacity due to the flexibility of the density family $\mathcal{Q}$. Variational inference allows us to play with different models and track their performance with many datasets. The disadvantage of variational inference is that its solution is only approximate and in some cases, we are not guaranteed to get close to good solutions. In contrast, the Gibbs sampling procedure will produce asymptotically exact samples of the posterior.

However, the disadvantage of Gibbs sampling (or MCMC in general) is the computational issue; it often takes a long time to access to the right area of samples. In contrast, variational inference relies on an optimization procedure, and, therefore, it can take advantage of modern stochastic techniques. Example techniques include stochastic variational inference [Hoffman et al., 2013] and the MapReduce distributed computing framework [Zhai et al., 2012]. Hence, VI solutions can scale to massive datasets and are more suitable with a diverse set of big datasets.

In this study's subsequent chapters, we use Gibbs sampling to recover the topic assignment (what proportions of topics are in each document or the document-topic matrix) because the anchor word algorithms (in Section 2.6 and in all subsequent chapters) only produce the topics (the topic-word distributions or the topic-word matrix).

2.3 Topic Model Evaluation

Latent topics discovered by topic models help users capture themes and explore corpus of interest. Hence, topics are often directly or indirectly used by researchers to compare different topic models, either though qualitative analysis or some form of ad-hoc quantitative analysis. In this section, we describe two popular metrics for evaluating topic models. The first metric evaluates the general goodness of a model (a predictive power) based on an unseen set of documents or *held-out documents*. This metric goes under the name *document held-out likelihood* and is a general machine learning evaluation metric for measuring generalizability of probabilistic models (how well the models generalize from unseen data). The second metric, called *topic interpretability*, evaluates the quality of topics produced by topic models. These two metrics complement each other and give us a systematic way to evaluate the quality of topic models. However, as we will see later, these two metrics do not always agree with each other and they sometimes show opposite trends of performance [Chang et al., 2009]. In practical usage of topic models, it is important to realize that not all metrics are useful, except for the metric that is closely related to the task of interest. Hence, it is more important to focus on designing task-based metrics than using general metrics, such as likelihood-based measures [Chang et al., 2009].

2.3.1 Document Held-out Likelihood

In Bayesian statistics, the likelihood function is the conditional probability of the observed data given values of the model's parameters: $p(x \mid z)$. Many statistical models (especially ones from traditional statistics) focus on maximizing the likelihood function to find the best parameters that can maximally explain data [Dempster et al., 1977, Chow, 1984, Akaike, 1998, Blei and McAuliffe, 2007]. To be concrete, assuming there is a process described by the model (e.g., the generative process of LDA or the process of generating the next word when given another word in language modeling), then the best parameters found by using the maximizing the likelihood function would likewise maximize the likelihood that the generative process would produce the predicted data that we observed. *Generalizability* measures the ability of a predictive model to make a broad inference on unseen data. If we compute the likelihood quantity based on the held-out documents (unseen data) using the learned parameters and this quantity is high, then we conclude that the model generalizes for unseen data or it is generalizable.

So the likelihood quantity can be used to measure the generalizability of the model (or the quality of the learned parameters regardless of how these parameters are learned). In probabilistic models such as LDA, posterior maximization is often the method of choice due to the rich priors on the space of parameters. A posterior can be achieved by multiplying the likelihood and the prior probability: $p(z \,|\, x) \propto p(x \,|\, z) \times p(z)$.

Computing Held-out Likelihood When evaluating a topic model, we split a corpus into a training set of documents D_{train} and a test set of documents D_{test}. We train a topic model based on the training set of documents to learn a set of parameters $params$ that often best explain the training set. We then compute the document held-out likelihood score based on the test set of documents, defined as

$$L = \sum_{d=1}^{|D_{test}|} \log(p(document_d \,|\, params)).$$

The document held-out likelihood metric evaluates the goodness of a topic model: the higher the score, the better the model. This metric is used not only for evaluating topic models but also for evaluating other types of machine learning models such as document clustering and segmentation. Note that in many cases, people use *perplexity* $\propto \exp\{-L\}$

to measure the held-out likelihood value. And for perplexity, a lower value implies better models.

In the case of the LDA model, Section 2.1.2 gives a well-formed formula of the likelihood value computed on the training set of documents. However, the process of generating a document in the test set $document_d$ given the model parameters $params$ is stochastic, and the held-out likelihood cannot be computed efficiently. Wallach et al. [2009b], Buntine [2009] provide various sampling methods to compute document held-out likelihood values. Throughout this thesis, we use the LDAC package, a C implementation of LDA by Dave Blei, to compute the held-out likelihood of LDA-like models.

2.3.2 Topic Interpretability Metric

Besides evaluating the generalizability of a topic model, people are also interested in evaluating the quality of generated topics. There are many ways to evaluate topic quality, either manually by human users or automatically by automatic tools. Using a human to evaluate topics gives the most accurate results. However, this approach is not scalable. Recently, people have found that using topic interpretability metrics is more appropriate in two aspects: they are easy to compute, and they also correlate well with human judgment of topics. Below, we present several common ways to measure topic interpretability.

Chang et al. [2009] propose using a word intrusion task to measure topic interpretability. A word intrusion task is the following: given a set of words (e.g., the top ten words from a topic), then randomly insert a word into that set, and then ask human users to spot that *intruder word*. If users perform well in this task, it would mean that the topic is interpretable. Otherwise, the topic is not interpretable. To run a large scale study, the authors used the Amazon Mechanical Turk for this task. Interestingly, they found that topic interpretability scores and document held-out likelihood values did not agree with each other.

Working with big data requires more scalable methods to evaluate topic quality. Most recent research studies have focused on inventing automatic techniques for measuring topic interpretability. Because the meaning of topics is visually captured by the top words from the topic-word distribution, one straightforward approach is to directly evaluate a topic through the interpretability score based on its top-N words. In addition, topic models learn topics using deeper word associations (second- or third-order)—a process that is very similar to how human's perception of a concept comes from a collective understanding of knowledge from many sources. Hence, comes many metrics that take advantage of second-order word associations: the word-word co-occurrences.

Newman et al. [2009, 2010b] propose pointwise mutual information (PMI) as the base measure for word-word co-occurrences to evaluate topic models. They used considerable corpora such as Wikipedia to generate PMI statistics based on how words co-occur together. Closely related to this approach, Mimno et al. [2011] introduce a log conditional probability as a replacement for PMI. Furthermore, they used this interpretability measure to directly improve the topic quality by incorporating words' relatedness counts into the sampling of a new topic assignment in the Gibbs sampler.

Aletras and Stevenson [2013] use distributional semantic similarity measures based on Wikipedia to compute the interpretability score of a topic. Because in the distributional hypothesis (see Section 1.2), each word is represented as a vector, it is easy to compute the similarity between a pair of words. Aletras and Stevenson [2013] measure interpretability of the top N-words of a topic by averaging scores of word pairs using Cosine similarity or a Dice coefficient. This approach is generic and can be combined with recent word-embedding models such as Word2Vec [Mikolov et al., 2013b] or Glove [Pennington et al., 2014].

Computing Pointwise Mutual Information

We review the basic formula of topic interpretability using PMI in Newman et al. [2010b]. The PMI of every pairs of words w_i and w_j is computed as

$$\text{PMI}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i) \times p(w_j)},$$

where $p(w)$ denotes the probability of observing the word w and $p(w_i, w_j)$ denotes the probability of observing the word w_i and the word w_j appearing within a same document. Sometimes, the context of two words can be changed, for example, when two words should not be too far away from each other within a document (window size).

To directly use PMI to measure topic interpretability of a topic k using top N-words, Newman et al. [2010b] define the observed coherence as

$$\text{OC-PMI}(k) = \sum_{j=2}^{N} \sum_{i=1}^{j-1} \text{PMI}(w_i, w_j).$$

In this thesis, we use the normalized PMI (hence, NPMI), defined as

$$\text{OC-NPMI}(k) = \sum_{j=2}^{N} \sum_{i=1}^{j-1} \frac{\text{PMI}(w_i, w_j)}{-\log(p(w_i, w_j))}.$$

The advantage of using NPMI (normalized pointwise mutual information) is that its values range from minus one to one so it is more intuitive to analyze and compare different topic models.

2.3.3 Task-Based and Other Evaluation Metrics

In the previous two sections, we have presented two intrinsic and automatic metrics to evaluate unsupervised topic models. Using these metrics in practice, however, should be limited to the initial evaluation of topic models; instead, a better metric should be built based on the task in which the topic models are used. Because applications of topic models vary a great deal, the main metrics depend on application domains. Examples include extrinsic evaluation metrics based on specific tasks such as document classification [Lu et al., 2011, Xie and Xing, 2013], improving information retrieval quality [Wei and Croft, 2006], or establishing a new label set [Poursabzi-Sangdeh et al., 2016]. To be concrete, for example, if the task is to classify documents using topic features, then it is better to use classification accuracy as the main metric and the topic interpretability as the secondary metric. We apply the strategy here in Chapter 4 to evaluate topic models for sentiment analysis task.

Another way to evaluate topic models that is used a lot by researchers and modelers is to measure the topic model performance on the synthetic data—data that is created by the generative process of LDA [Wallach et al., 2009b, Alsumait et al., 2009, Andrzejewski et al., 2009a, Taddy, 2012, Arora et al., 2013]. To measure a topic model, we compare the topics from synthetic data with topics learned by the model. In addition, we can compare document-topic distributions (topic assignment) from synthetic data with those produced by the model; similarly, we can visually inspect topics produced by the model.

2.4 LDA Extensions

Since the introduction of LDA, there have been numerous extensions that try to address different modeling questions as well as application needs. In this section, we review several extensions of LDA that motivate the works in this thesis. These extensions include improving scalability, adding supervision, and enriching models for external knowledge. Also, we provide a brief overview of other types of topic models that are useful for research purposes and practical applications.

2.4.1 Scaling Up Topic Models

The unsupervised LDA model and its subsequent models have shown important applications in understanding text, and are useful for many downstream tasks such as text classification or regression [Wang and McCallum, 2006, Blei and McAuliffe, 2007, Lacoste-Julien et al., 2009, Ramage et al., 2009, Zhu et al., 2009a, Ramage et al., 2011]. However, these models often suffer from sluggish performance on large datasets due to the slow inference algorithms such as Gibbs sampling or variational inference (see Section 2.2). Because of this, scaling up topic models for big datasets has been addressed intensively in recent years.

The first set of ideas directly addresses the inference steps used by topic models. For example, Smola and Narayanamurthy [2010] build a parallel sampling architecture for the inference of topic models across many clusters while Nallapati et al. [2007] build a parallel version of variational inference. Similarly, Wang et al. [2009c] create a parallel version of LDA using message passing interface (MPI) and MapReduce, Zhai et al. [2012] implement variational inference for LDA in MapReduce, or Huang et al. [2014a] propose using tensor factorization to build a distributed version for LDA. The common theme in these approaches has been software and hardware architectures to obviate the slow synchronization process for the topic model algorithms (Gibbs sampling or variational inference) [Asuncion et al., 2008]. This line of research has the advantage of simplicity and is straightforward to implement. Another advantage is that as new parallel architectures come into the market, an improved version of LDA inference will arrive. Recently, Spark framework has drastically helped to boost the usage of the LDA model in many practical big data applications.[4]

Another research direction for large scale applications of topic models is online learning [Wang et al., 2011, Wang and Blei, 2012, Bryant and Sudderth, 2012, Li et al., 2014, Hoffman et al., 2013]. Online learning for topic models takes advantage of the way variational inference reduces computation costs by truncating dependencies among latent variables to avoid multiple passes through data. The general idea is to optimize the variational objective function of topic models using stochastic optimization algorithms. The stochastic algorithms only use a random subset of data (e.g., documents) to update

[4] https://spark.apache.org/

the variational parameters for this subset. Online topic models are useful for massive and streaming datasets such (e.g., social news).

Finally, the anchor word algorithm by Arora et al. [2013] (details described in Section 2.6) and its subsequent extensions introduced in this thesis provide very fast and scalable solutions for topic models. The core idea of anchor word algorithms is the usage of a separability assumption [Donoho and Stodden, 2004] to turn a non-convex non-negative matrix factorization problem into solving convex optimization problems, hence, reducing problem complexity and improving computation costs.

2.4.2 Adding Supervision

Unsupervised learning models such as LDA help us understand the general theme for large document collections. However, because these models are generative models, they may not be the best candidates (compared to discriminative classifiers/regressors) for addressing tasks such as prediction [Ng and Jordan, 2001]. Application of LDA to prediction is limited because LDA only provides the low dimensional representation for documents as vectors of latent topics—which is pretty much like common dimensionality reduction techniques. Compared to standard supervised machine learning techniques such as support vector machine or logistic regression based on word n-gram features, LDA often performs worse.

It is possible to extend unsupervised learning models to incorporate metadata information. The idea is to build hybrid models that can both understand the landscape of data (e.g., themes or topics) and make the correct prediction. A new class of learning models emerges under the name of *supervised topic models*. A supervised topic model is an extension of an unsupervised topic model that often concerns with different types of labels or metadata. In these subsequent paragraphs, we introduce several recent supervised topic models that we find useful for research as well as practical applications.

The first class of extensions focuses on directly extending LDA for the prediction task, primarily for regression and classification. Typically, in this category, Blei and McAuliffe [2007] introduce supervised LDA (hence, SLDA) to predict regression or labels from documents. In SLDA, a label (or a regressed value) is assumed to be generated from the observed topic assignment of words within each document. Meanwhile, Lacoste-Julien et al. [2009] introduce the discriminative LDA (DisLDA) for classification. DisLDA uses a single class to modify the document specific topic proportion by applying a class-dependent linear transformation. To deal with multi-labeled documents where each document is tagged with multiple labels, Ramage et al. [2009] introduce the labeled-LDA that assumes a one-on-one association between each label and each topic. In the labeled-LDA, the word token in each document can only be generated from those topics that are associated with observed labels of the document. Subsequent work by Ramage et al. [2011] extend the labeled-LDA to create partially labeled-LDA for text mining. The combination of LDA and an SVM-like classifier also produce a powerful model. For example, Zhu et al. [2009a] introduce a max-margin topic model for both regression and classification that boosts up the predictive power of LDA while retaining the quality of produced topics.

Going beyond labels and regression, another category of models addresses several types of metadata. Examples of these models include the work by Erosheva et al. [2004] that capture references, the work by Wang and McCallum [2006] that capture timestamps, the work by Nallapati et al. [2008] for jointly capturing texts and citations, or the work by Newman et al. [2006] that capture entities with topics. The general solution in this category is to represent metadata as additional dimensions to the word representations (e.g., the topic-word distributions β). Our supervised anchor word algorithm in Chapter 4 falls under this category. Specifically, we extend the anchor word algorithm [Arora et al., 2013] to incorporate supervision by enriching the vector representation for words by incorporating word-metadata co-occurrence statistics.

The third category of models concerns a more complex relationship (often by co-variates) among metadata or labels. Mimno and McCallum [2007] propose the author-persona-topic model to capture dependencies among complex authorship information. Similarly, Rubin et al. [2012] create the dependency-LDA and the prior-LDA models to capture the dependencies among labels by projecting them onto a lower dimensional latent space.

Other topic models capture intrinsic nuances within and across languages. These types of models are useful for absorbing multi-lingual metadata across multi-lingual documents (e.g., metadata is the languages of documents). The first example is the multilingual topic model (MTM) [Boyd-Graber and Blei, 2009, Boyd-Graber and Resnik, 2010]. MTM uses aligned text from different languages to learn topics that are useful for a single language as well as for capturing cross-lingual topics. Similarly, Mimno et al. [2009], Hu et al. [2014b] extend LDA to learn aligned topics across languages. In some cases, these models do not need aligned datasets, but utilize comparable texts to extract topics across languages [Jagarlamudi and Daumé III, 2010]. Recently, Yuan et al. [2018] introduce the multilingual topic anchors (MTAnchor) that use the anchor topic model instead of LDA to capture topics in English, Chinese, and Sinhalese documents. One advantage of MTAnchor in comparison to MTM is its speed. MTAnchor also supports user interaction and refinement of topics.

Overview of Supervised LDA To date, supervised LDA is still the most popular supervised topic model. That is partially due to the generative process of SLDA that is very much like that of LDA. The SLDA generative process only adds one additional step that models how the response variable (labels or regression values) is generated after topics for every word in a document have been assigned.

We describe the generative process of SLDA below, using the same notations as used in the generative process of LDA.

1. Draw K topic over words vectors: $\beta_k = \text{Dir}_V(\boldsymbol{\eta})$
2. For each document $d = 1, \ldots, D$ do
    - Draw a document length: $N_d = \text{Poison}(N)$
    - Draw a topic proportions: $\theta_d = \text{Dir}_K(\boldsymbol{\alpha})$
    - For each word $n = 1, \ldots, N_d$ do
        - Draw a topic assignment: $z_{d,n} = \text{Multinomial}(\theta_d)$, $z_{d,n} \in [K]$.
        - Draw a word: $w_{d,n} = \text{Multinomial}(\beta_{z_{d,n}})$, $w_{d,n} \in [V]$.
    - Draw a response $y_d \sim \boldsymbol{N}(\zeta^T \bar{z}_d, \delta)$ where $\bar{z}_{d,k} = \frac{1}{N_d} \sum_{n=1}^{N_d} \mathbb{1}[z_{d,n} = k]$.

Where we define $\mathbb{1}[x] = 1$ if x is true, and 0 otherwise.

In SLDA, the response variable is modeled as the normal distribution of the *empirical topic assignment*. Words and responses are not generated by topics since documents are generated first and only then comes the response variable. Because of the similarity between LDA and SLDA, we can use the inference techniques discussed in Section 2.2 to find the solution for SLDA. Compared to LDA, the SLDA inference algorithms have additional complication that comes from the added response variable. For example, according to Blei and McAuliffe [2007], the variational update on the variational parameters ϕ_j depends on the variational parameters ϕ_{-j} of all other words, therefore the ϕ_j cannot be updated in parallel [Blei and McAuliffe, 2007]. Hence, the SLDA model often runs much slower than LDA, and it is often trickier to scale up.

Direct extensions of the SLDA model include the multilingual supervised LDA [Boyd-Graber and Resnik, 2010], the max-margin supervised LDA [Zhu et al., 2009a], the hierarchical relation model [Chang and Blei, 2010] for modeling networks of documents and links, and the multi-class supervised LDA [Wang et al., 2009a] for capturing multiple classes and annotations with application to image classification. Moreover, recently, Card et al. [2018] combine SLDA with sparse additive generative models (SAGE) [Eisenstein et al., 2011] to create a neural model that can incorporate prior knowledge in the form of embeddings while still capturing covariates among topics and labels.

2.4.3 Incorporating Domain Knowledge into Topic Models

Topic models consume large text documents and produce semantic representation through topics. However, information fed to topic models not only includes documents but also contains side information such as metadata or expert knowledge. Topic models will become more useful if they can capture this additional information. Also, with this ability topic models will learn more extensive topics and adapt better to user needs. One concrete example is to use topic modeling for understanding biomedical text such as on PubMed. Using LDA will only produce topics about general diseases or drugs that lack the domain knowledge about how diseases and drugs connect. The ability to embed medical ontology data into topic models will capture these associations. We address this issue of incorporating domain knowledge in Chapter 3 where we embed the LIWC (Linguistic Inquiry and Word Count) knowledge [Pennebaker and Francis, 1999] into the anchor word algorithm to create more interpretable topics.

The need to incorporate external information into topic models becomes more important when we work with small datasets that do not contain enough signals. To understand these small datasets, we often have to convey expert knowledge about how terms or topics should go together. In this situation, domain knowledge becomes very valuable to produce interpretable topics and meanings. In this section, we describe two classes of topic modeling extensions that capture two favorite types of textual domain knowledge.

The first type of knowledge that we want topic models to capture is the composition of words. To be precise, we want certain words to appear on specific topics. Boyd-Graber et al. [2007] introduce a latent Dirichlet allocation with WORDNET (LDAWN) to address the issue of word sense disambiguation by extending LDA to incorporate WORDNET [Fellbaum, 2005]. LDAWN introduces word senses as latent variables into LDA, hence, it can learn the context in which a word is disambiguated, producing topics as well as assigning specific meaning to each word. Subsequently, Andrzejewski et al. [2009b] propose a general solution for incorporating domain knowledge where they define prior knowledge as probabilities of the composition of words within topics. By representing the composition of words through *Must-Links* and *Cannot-Links* word pairs, they used Dirichlet Forest prior to replace the Dirichlet prior over the topic-word distribution in LDA. Their approach can capture the strength of links, and also yield fast inference using collapsed Gibbs sampling.

Another interesting work is the *Fold-All* model by Andrzejewski et al. [2011]. Fold-All represents user prior knowledge as first-order logic rules and then converts these rules into a Markov random field. Combining the produced Markov random field with LDA, Fold-All can be efficiently learned using stochastic gradient descent. Similarly, Jagarlamudi et al. [2012] use the concept of *seed words* to mean a list of words that must belong to a *seed* topic. They address this problem by directly incorporating a Bernoulli distribution for saying whether a word belongs to the seed topic or not. This approach enforces related words to come together within one seed topic by penalizing the distribution of those words across topics [Jagarlamudi et al., 2012].

The second class of prior knowledge involves more complex relationships among words and documents. For example, Yang and Leskovec [2015] introduce sparse constrained LDA that incorporates word correlation knowledge and document label knowledge. Similarly, Xie et al. [2015b] build a model for incorporating word correlation using a Markov random field over latent topics.

In the sense that linguistic properties can also improve topic models, Griffiths et al. [2005] relax the bag-of-words assumption in LDA to integrate topic and linguistic syntax information. Subsequent works address n-grams [Wallach, 2006], phrases [Wang et al., 2007a, Lindsey et al., 2012], and parse trees [Boyd-Graber and Blei, 2008].

By allowing users to refine the topics produced by topic models interactively, the interactive topic modeling framework [Hu et al., 2014a] mitigates the incorporation of human knowledge and improves topic quality. The rise of deep learning architectures and algorithms also produce some deep topic models [Hinton and Salakhutdinov, 2009, Larochelle and Lauly, 2012]. In Chapter 5, we introduce the combination of active learning and supervised topic models for classification problems.

2.5 Spectral Methods for Topic Models

In parallel with probabilistic approaches, algebraic approaches have also been developed for topic models. In this realm, *spectral methods* are the most popular. Spectral methods include algorithms that rely on algebraic operators such as computing eigenvalues and eigenvectors. Well-known examples are the singular-value decomposition (SVD) and the principal component analysis (PCA) [Pearson, 1901, Jolliffe, 1986]. So what exactly are spectral methods? Also, how do we distinguish them from other machine learning techniques?

According to Shizgal [2015],

> The origin of the term, *spectral* is not entirely clear but probably arises from the original use of Fourier sines and cosines as basis functions (Gottlieb and Orszag 1977; Brown and Churchill 1993) especially in connection with a time series analysis and the fundamental frequencies of a process, namely the

spectrum (Shen et al. 2011).

So spectral methods are techniques that can decompose any signal or data into a summation of basic factors (called *bases*). From a machine learning perspective, spectral methods are techniques or algorithms that decompose data statistics into explainable parts. For example, Lee and Seung [1999] use non-negative matrix factorization (NMF) to learn parts of faces and semantic features of textual content. So, NMF is a spectral method. A few other techniques include principal component analysis [Pearson, 1901, Jolliffe, 1986], vector quantization (VQ), parallel factor analysis [Harshman and Lundy, 1994], or factorization machines for recommendation systems [Rendle, 2010, Bell et al., 2009]. Each class of algorithms differs in how we place constraints on parts when performing decomposition.

Recently, spectral methods have found their way into topic models. Spectral methods provide a natural solution because latent variable models such as topic models focus on learning a small number of latent unobserved variables (e.g., topics) to explain a large number of observed variables (e.g., documents and words). Examples include provable guarantees solutions using anchor methods [Cohen et al., 2013, Arora et al., 2013, Foster et al., 2012] or moment methods [Anandkumar et al., 2012c,a, Huang et al., 2015]. Unlike variational EM algorithms that only provide local approximations for topic models (Section 2.2.2), under certain assumptions, these spectral methods can recover global solutions for topic models. For example, the anchor word algorithm in Arora et al. [2013] assumes a separability assumption in which factors are assumed to be part of the observed signals, simplifying the solution drastically. Similarly, Anandkumar et al. [2012a]

introduce excess correlation analysis (ECA) that use spectral decomposition to learn latent topics (factors) from third-order moments of words. Both of these techniques are scalable and provide global solutions for topic models.

Historically, spectral methods for topic models are not new. Before LDA, the seminal work of latent semantic analysis (LSA) by Deerwester et al. [1990], Papadimitriou et al. [1998] initiates solutions to simpler versions of LDA. The core idea of LSA is the application of singular vector decomposition (SVD) to factorize the document-word matrix for the document retrieval task. Because SVD reduces the dependent dimensions using orthogonal factors, each factor will correspond to a topic in topic modeling sense. The result of LSA is that each document is projected as a vector of topics in lower dimensional space, which then can be used to compare documents across the corpus for the information retrieval task.

We can learn topic models by factorizing the document-word using non-negative matrix factorization. The trick is to place non-negativity constraints on the document-topic and the topic-word factor matrices. In the next section, we formally describe NMF for solving topic models.

2.5.1 Non-negative Matrix Factorization

Non-negative matrix factorization [Lee and Seung, 1999, 2001, Donoho and Stodden, 2004] is a general method for learning the latent structure of a matrix where elements on factored matrices are non-negative. NMF has an application in many fields, including document clustering [Ding et al., 2005], system biology [Karthik, 2008], music analysis [Févotte

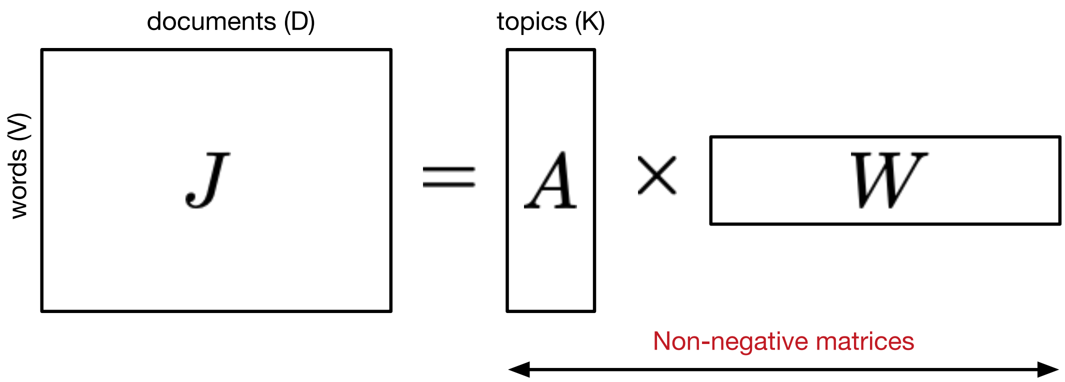


Figure 2.3: An illustration for an NMF problem where we factorize the matrix J into two non-negative matrices A and W.

et al., 2009], recommendation system [Gemulla et al., 2011], and topic models [Arora et al., 2012a, 2013, Anandkumar et al., 2012a, Huang et al., 2015].

The non-negative solutions of non-negative matrix factorization are suitable for topic models because topic models deal with probabilities. A well-known application of NMF is to factorize the document-word matrix where the factorized document-topic matrix and topic-word matrix have non-negative elements. We will use the language of topic modeling and some notations in Section 2.1.2 to describe the NMF problem.

We will work with a corpus of D documents and V words. Our goal is to factorize the term-document matrix J of size V by D where each cell $J_{v,d}$ is the frequency of the word v in the document d. Hence, each column of the matrix J corresponds to a document in the corpus; it is a sparse vector of word counts. NMF finds latent structures (e.g., topics) by factorizing the matrix J into two matrices A of size V by K and W of size K by D: $J \approx AW$. The non-negativity constraints requires $A \geq 0$ and $W \geq 0$. Figure 2.3 illustrates this problem.

To find the solution, we are looking to solve the objective:

$$\text{argmin}_{A\geq 0, W\geq 0}|J - AW|,$$

using some distance metrics such as the KL divergence or the L_2 loss.[5]

NMF using Kullback-Leibler Divergence

Using KL divergence, the objective function becomes

$$\text{argmin}_{A\geq 0, W\geq 0}\sum_{v}^{V}\sum_{d}^{D} J_{v,d}\log(\frac{J_{v,d}}{(AW)_{v,d}}) - J_{v,d} + (AW)_{v,d}. \tag{2.23}$$

Ding et al. [2008] show that there is an equivalence between the above NMF problem and the probabilistic topic model PLSA: they share the same objective function given that a word v and a document d are conditionally independent given a topic k. To understand this, let revise the objective function of PLSA [Hofmann, 1999b] where it maximizes the likelihood:

$$\text{argmax}\sum_{v}^{V}\sum_{d}^{D} J_{v,d}\log(p(word = v, doc = d)), \tag{2.24}$$

where $p(word = v, doc = d)$ is the joint probability of a word and a document, we have

$$p(word = v, doc = d) = \sum_{k=1}^{K} p(word = v\,|\,topic = k)p(doc = d\,|\,topic = k)p(topic = k).$$

[5] The final NMF solutions for topic models often require us to perform a column normalization operation on the matrix A and the matrix W.

Rewrite Equation 2.24, the object function

$$
\begin{aligned}
&\doteq \operatorname{argmax} \sum_{v}^{V} \sum_{d}^{D} J_{v,d} \log(p(word = v, doc = d)) \\
&\equiv \operatorname{argmin} \sum_{v}^{V} \sum_{d}^{D} -J_{v,d} \log(p(word = v, doc = d)) \\
&\equiv \operatorname{argmin} \sum_{v}^{V} \sum_{d}^{D} J_{v,d} \log(\frac{J_{v,d}}{p(word = v, doc = d)}) \\
&\equiv \operatorname{argmin} \sum_{v}^{V} \sum_{d}^{D} J_{v,d} \log(\frac{J_{v,d}}{p(word = v, doc = d)}) - J_{v,d} + p(word = v, doc = d).
\end{aligned}
\tag{2.25}
$$

The last formula of the above Equation 2.25 looks exactly like Equation 2.23 of NMF using KL divergence.

Recently, Faleiros and Lopes [2016] demonstrate that the objective function of NMF using KL divergence (Equation 2.23) approximates the variational inference algorithm for LDA in which we maximize the ELBO function (Equation 2.17).

Sparsity of NMF Solutions

Due to the nature of the optimization problem, using NMF often results in sparse solution matrices [Ding et al., 2010]. This property is beneficial for topic models because it mimics the Dirichlet priors in the LDA model in which only a small number of topics are available in each document, and only a small number of words are used to explain topics.

To understand why it is the case, let us take the derivative of the objective function using L_2 loss:

$$f(A, W) = ||J - AW||_F^2.$$

The first-order optimality conditions [Gillis, 2014] are

$$A \geq 0, \nabla_A f = AWW^T - JW^T \geq 0, A \circ \nabla_A f = 0,$$

$$W \geq 0, \nabla_W f = A^T AW - A^T J \geq 0, W \circ \nabla_W f = 0,$$

where $\circ$ denotes the component-wise product of two matrices. The above conditions suggest that the solutions of NMF will be sparse because values are either zeros or positive (products with gradient).

Common Issues of NMF Because NMF is a constrained optimization problem on matrices, it has some issues that prevent us from finding the solution quickly.

The first issue is that the NMF problem is NP-hard [Vavasis, 2009]. Finding an exact solution for NMF is intractable [Arora et al., 2012a, Moitra, 2013]. To overcome this issue people find approximate solutions. A simple approach is to apply quick techniques like SVD or QR to factorize the matrix J. This will result in factors that have both negative and positive values, but we will then replace those negative values with zeros. Another approach is to place some assumptions on the solution matrices A or W to speed up computation. One typical assumption is the separability assumption by Donoho and Stodden [2004]. Given this assumption, factors learned by NMF are assumed to come from the observed data, hence, the intractability issue is amended since the problem becomes convex. The anchor word algorithm and our extensions in this thesis use the separability assumption to devise a fast solution for topic models. We will describe more about this assumption in Section 2.6 where we discuss the anchor word algorithm.

The second issue with NMF is *rotation*. Given solutions A and W, if there exists a matrix R such as $AR \geq 0$ and $R^{-1}W \geq 0$, then these new matrices are also solutions for the same NMF problem. A general strategy to address this issue is to add more constraints on the solution matrices A and W. For example, Hoyer [2004], Gillis [2012], Kim and Park [2007] introduce sparsity constraints to select the sparsest solutions for A and W. Another solution is to do normalization on columns of A and W [Ding et al., 2008].

This thesis focuses on a particular class of NMF using the separability assumption called *separable-NMF*. With the separability assumption, separable-NMF is guaranteed to produce fast solutions for topic models. Specifically, if the rows of the matrix W are a subset of rows of the matrix J, then there would be a procedure for finding W quickly [Chan et al., 2008, Kumar et al., 2013, Bioucas-Dias et al., 2012]. After we have recovered W, finding A is easy because the problem becomes convex. Arora et al. [2012a, 2013] develop the anchor word algorithm using this idea for topic models and achieved impressive results. We will review the anchor word algorithm in Section 2.6, and introduce our solutions based on the anchor word algorithm to address issues of scalability, variability, and interactivity in the subsequent chapters.

2.5.2 Other Spectral Methods

In parallel with the NMF approach for topic modeling, researchers have pursued the general method of moments to estimate mixture models [Anandkumar et al., 2012a]. Anandkumar et al. [2014] further extend this direction and introduce a general tensor decomposition framework for latent variable models. Ideas from this line of research can be used to estimate parameters of LDA using trigram statistics (e.g., third-order moments) [Anandkumar et al., 2012b,a].

Very similar to the anchor word algorithm, Ding et al. [2013a,b] propose an approach that uses data-dependent and random projection to discover novel words for topics. In their approach, each word is represented as a vector of length equal to the number of documents (the matrix J of size V by D) instead of using word-word co-occurrence as in the anchor word algorithm (see below). Using this representation, they then find cross-document patterns of words in lower dimensional space to select K novel words for K topics. Finally, they recover topics by exploiting the separability condition.

2.6 Unsupervised Anchor Word Topic Models

As discussed above, we focus on the separable-NMF in this section and show how it can provide a scalable solution for topic models. Given the document matrix J of words and documents (size V by D), we are looking for the matrix A of words and topics (size V by K) and the matrix W of topics and documents (size K by D) that satisfy $J \approx AW$.

Anchor Word Assumption Given K topics, we assume that there are at least K words (hence, called *anchor words*), one anchor word for one topic, so that every time we see an anchor word of a topic in a document we know that the document is partially about that topic.

Intuitively, this means that each topic has a unique, specific word that, when used, identifies that topic. For example, while “run”, “base”, “fly”, and “shortstop” are associated with a topic about baseball, only “shortstop” is unambiguous, so it could serve as this topic’s anchor word. This assumption is the separability assumption [Donoho and Stodden, 2004], but in topic modeling language.

The formal definition for the separability condition is restated in Arora et al. [2012b] as follows: *The topic matrix A of size V by K is called p-separable if for each i there is some row of A that has a single non-zero entry which is in the i^{th} column and its value is at least p.*

What property do we have when A is p-separable? Recall that, for any anchor word index g_k for the topic k: $A_{g_k,k} > 0$ and $A_{g_k,k^{'}} = 0$ for any $k^{'} \neq k$. We have

$$J_{g_k,\cdot} = \sum_{k=1}^{K} A_{g_k,k} W_{k,\cdot} = A_{g_k,k} W_{k,\cdot}.$$

From the algebraic formula, each row of the matrix W is a row of the matrix J with a multiplicative factor. We will do row-normalization of the matrix A later so that each row of W will correspond precisely to one row of J. The row indexes of J that correspond to rows of W are the anchor word indexes. So if we can find those anchor word indexes from J, we have established the matrix W without any computation. Subsequently, finding the matrix A will be a convex problem.

Issues with Using J The input word-document matrix J is very sparse because each document only uses a small number of words (compared to all the words in the vocabulary). Therefore, running matrix factorization on the matrix J, it is very hard to reliably recover A. Instead, we want to use a more statistically reliable input to recover topic matrix A. Arora et al. [2013] propose to use the matrix $Q \equiv JJ^T$ to recover A. The interpretation of Q is that it captures all co-occurrence of word pairs in every documents from the corpus;

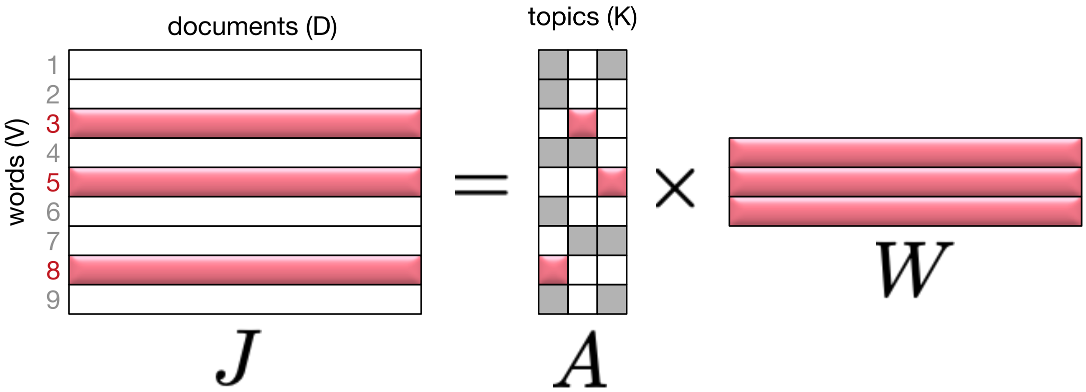


Figure 2.4: Illustration *separability condition*. In the matrix A, blank cells mean 0s, other cells (darker color) have values greater than 0. For each column k (topic) of the matrix A, there exist at least one row index i such that $A_{i,k} > 0$. In this figure, three anchor word indexes are $\{3, 5, 8\}$. Each row of the matrix W corresponds to one row in the matrix J.

Q contains co-occurrence statistics of every word pairs and has a dimension of V by V. The following equation shows that we can use the matrix Q to recover the matrix A:

$$Q \equiv JJ^T = AWW^TA^T. \tag{2.26}$$

A is still separable because the rows of the matrix WW^TA^T is the subset of the rows of the matrix JJ^T (with multiplicative factors). Hence, we can reliably apply the same separable-NMF algorithm using the word co-occurrence statistics Q as the input matrix.

In the next paragraph, we briefly review the anchor algorithm in Arora et al. [2013] where they use the word co-occurrence matrix Q to recover the topic matrix A.

Rethinking Data: Word Co-occurrence We represent the joint distribution of words as $Q_{i,j} = p(w_1 = i, w_2 = j)$. Each cell of the matrix Q is the probability of words appearing together in a document.

Let us assume that we knew what the anchor words were: a set $\mathcal{G}$ that indexes rows in the matrix Q. Now consider the *conditional distribution* of word i, the probability of the rest of the vocabulary given an observation of word i; we represent this as $\bar{\boldsymbol{Q}}_{i,\cdot}$, as we can construct this by normalizing the rows of Q. For an anchor word $g_k \in \mathcal{G}$, this will look like a topic; $\bar{\boldsymbol{Q}}_{\text{“shortstop”},\cdot}$ will have high probability for words associated with <u>baseball</u>.

The critical insight of the anchor algorithm is that the conditional distribution of non-anchor words can be reconstructed as a linear combination of the conditional distributions of anchor words. For example, $\bar{\boldsymbol{Q}}_{\text{“fly”},\cdot}$ could be reconstructed by combining the anchor words “insecta”, “boeing”, and “shortshop” (Figure 2.5). We represent the coefficients of this reconstruction as a matrix C, where $C_{i,k} = p(z = k \,|\, w = i)$. Thus, for any word i,

$$\bar{\boldsymbol{Q}}_{i,\cdot} \approx \sum_{g_k \in \mathcal{G}} C_{i,k} \bar{\boldsymbol{Q}}_{g_k,\cdot}. \tag{2.27}$$

The coefficient matrix C is **not** the usual output of a topic modeling algorithm. The normal output is the matrix A of the probability of a word *given a topic* while the coefficient matrix C is the probability of a topic *given a word*. We can think of C as the row-normalized A. After having C, we use Bayes rule to recover the topic distribution $p(w = i \,|\, z = k) \equiv$

$$\begin{aligned} A_{i,k} &\propto p(z = k \,|\, w = i) p(w = i) \\ &= C_{i,k} \sum_j \bar{Q}_{i,j} \end{aligned} \tag{2.28}$$

where $p(w)$ is the normalizer of Q to obtain $\bar{\boldsymbol{Q}}_{w,\cdot}$. Figure 2.6 shows how the anchor word algorithm works; it uses anchor words and the Q matrix to recover topics.

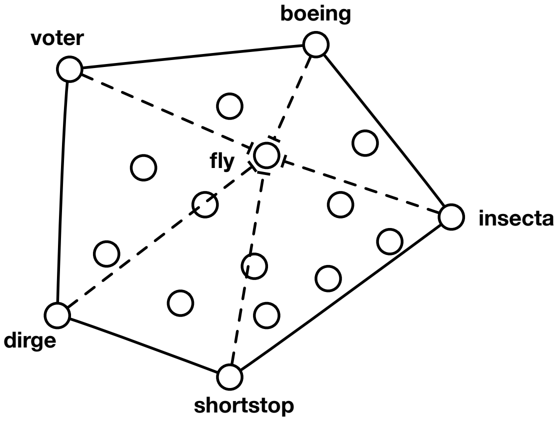


Figure 2.5: Geometric intuition behind the anchor word algorithm, each word is represented as a vector of word co-occurrence with all vocabulary. Anchor words such as “insecta”, “boeing”, “shortstop”, “voter”, and “dirge” form the convex hull of word co-occurrence probabilities. A ambiguous word such as “fly” stays inside and is a linear combination of anchor words.

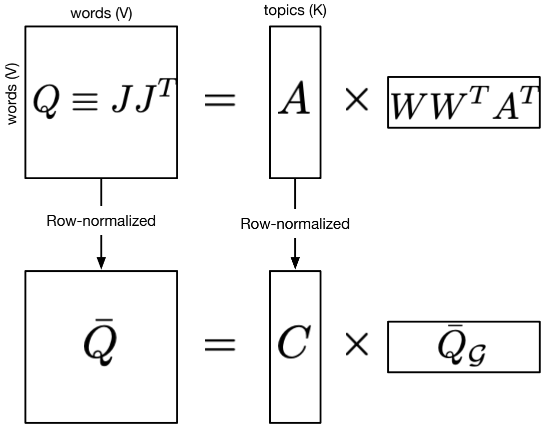


Figure 2.6: Illustration of an NMF problem to recover the topic coefficients C from the word-word co-occurrence statistics Q. Given a set of anchor word indexes $\mathcal{G}$, we can easily find each row of C by solving a convex problem.

Finding Anchor Words The geometric argument of the anchor word algorithm is illustrated in the Figure 2.5. In this figure, each word, which corresponds to a row of the matrix $\bar{Q}$, is a point in a high dimensional space. The key idea is that each anchor word will be vertices of a convex hull where every other word stays inside; finding anchor words is equal to finding these vertices. Arora et al. [2013] propose an approach called *FastAnchorWords* in which it iteratively finds the furthest point from the subspace spanned by previous anchor words. For completeness, we describe in detail this algorithm in Appendix A, Section A.5. We keep this procedure unchanged and use it throughout this thesis. Also, to produce the high quality anchor words, we use a document frequency threshold value M to filter anchor word candidates that appear in less than M documents.

From Anchors to Topics After we have the anchor words, we need to find the coefficients that best reconstruct the data $\bar{Q}$ (Equation 2.27). Chosing the C that minimizes the KL divergence between $\bar{Q}_{i,\cdot}$ and the reconstruction based on the anchor word's conditional word vectors $\sum_{g_k \in \mathcal{G}} C_{i,k} \bar{\boldsymbol{Q}}_{g_k,\cdot}$,

$$\boldsymbol{C}_{i,\cdot} = \text{argmin}_{\boldsymbol{C}_{i,\cdot}} D_{\text{KL}} \left(\bar{\boldsymbol{Q}}_{i,\cdot} \,||\, \sum_{g_k \in \mathcal{G}} C_{i,k} \bar{\boldsymbol{Q}}_{g_k,\cdot} \right). \tag{2.29}$$

The above Equation 2.29 corresponds to the NMF algorithm using KL divergence. In addition, because of the separability assumption (anchor word assumption), we know the K vectors $\bar{\boldsymbol{Q}}_{g_k,\cdot}$s, hence, recovering C becomes a convex optimization problem. Therefore, the anchor word algorithm is fast, as it only depends on the size of the vocabulary once the co-occurrence statistics Q are obtained. Equation 2.29 is actually similar to Equation 2.26, the only difference between them is that we row-normalize the matrix Q (to become $\bar{Q}$ and we row-normalize the matrix A to become C (Figure 2.6).

The anchor algorithm is deterministic in a sense that it will recover the same topic models given a fixed set of anchor words. It is different from probabilistic inference where we may get different topics for different runs.

The existence of anchor words corresponds to the separability assumption for NMF,

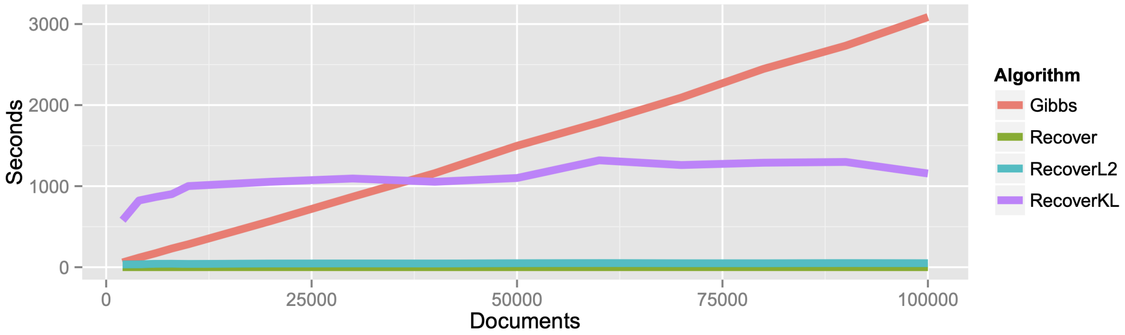


Figure 2.7: Benchmarking using synthetic Neural Information Processing Systems documents from Figure 1 in Arora et al. [2013]. RecoverKL (Equation 2.29) is the algorithm we use throughout all chapters in this thesis. As the figure shows, Gibbs sampling is linear in the corpus size while RecoverKL takes a constant time. The efficiency of the anchor word algorithms starts to show when the number of documents reaches around 37,000. Note that this benchmark was conducted with a non-distributed implementation of the anchor word algorithm. In this thesis, we use a distributed implementation of the anchor word algorithm, which still takes a much less constant time. For example, running on a ten-core machine, we expect the training time is roughly 100 seconds and the efficiency of the anchor word algorithm shows when the number of documents reaches 3,700.

and it has been used as a key property to recover topic models quickly. Under this assumption, we assume that each topic has at least one anchor word. Recently, Ding et al. [2015] show that given a sufficiently large corpus, separability is an inevitable result of high-dimensionality. Especially, this result ties up with the fact that topic models assume two Dirichlet priors over document-topic and topic-word distributions. In practice, however, using anchor topic models require us to verify this assumption carefully by observing data statistics as well as by empirically evaluating the quality of topics produced by them using topic interpretability as well as task-based metrics.

2.6.1 Anchor Method is faster than Gibbs & VI

In this section, we give a brief analysis of the computational complexity of the anchor algorithm in comparison to the Gibbs sampling and the variational inference.

From the formulation of the anchor word algorithm described above, the only corpus statistics that it uses to reconstruct topics is the word-word co-occurrence matrix Q. Forming Q takes roughly $O(D \times N^2)$ where N is the expected length of a document and D is the size of the corpus. This preprocessing step can be done in parallel. Finding anchor words using FastAnchorWords procedure (Section A.5) takes $O(V^2)$, even though this can be reduced using document threshold M. After forming the co-occurrence matrix Q once, the anchor word algorithm invokes V optimization solvers for every word term. Also, these solvers are independent of each other, so we can solve them independently. Consequently, the reconstruction step of the anchor word algorithm (Equation 2.29) only relies on the number of words (roughly $O(K \times V^2)$ [Arora et al., 2013]) which is very small compared to the number of documents.

In contrast, the runtime of Gibbs sampling and variational inference for LDA depends on the number of documents and the number of words in each document. As a result, Gibbs sampling and variational inference will perform much worse than the anchor word algorithm for large datasets. Running the anchor algorithm is like performing one Gibbs sampling iteration.

Figure 2.7 demonstrates how efficient the anchor word algorithms compared to Gibbs sampling. This figure (we reuse the Figure 1 from Arora et al. [2013]) compares the training time of Gibbs sampling with the training time of various anchor word algorithms on synthetic Neural Information Processing Systems documents. RecoverKL corresponds to the anchor word algorithm using KL divergence as an objective (Equation 2.29).

2.6.2 Contributions

The anchor word algorithm provides a quick solution for topic models; its runtime is linear to the number of word types in the vocabulary (except for the preprocessing step to construct Q, which depends on the number of documents—but can be done in parallel). In this thesis, we contribute by using standard topic evaluation metrics to evaluate the anchor word algorithm performance against the Gibbs sampling and the variational inference. In Chapter 3, we thoroughly evaluate the quality of topics using topic interpretability and the general quality of the models using document held-out likelihood.

The original anchor word algorithm has not been used to handle various types of datasets (e.g., adding external knowledge or adding metadata); hence, it is not flexible to address variability. We contribute to the literature by addressing the issue of data variability and external knowledge in Chapter 3 where we incorporate informed priors by using regularization. In addition, in Chapter 4 we enrich the representation of words using metadata (e.g., labels) to help discover new latent topics that explain sentiment datasets.

One problem with the original anchor word algorithm is that we can only recover the topic matrix A but not the document topic matrix W. This problem is due to the stochastic nature of how documents are generated. This is why we still have to convey Gibbs sampling to recover the document-topic matrix, especially in the case where we need those features for classification or document retrieval applications. We introduce a fix for this issue in the context of active learning for classification in Chapter 5.

The separability assumption used by the anchor word algorithm helps to create a fast algorithm for topic modeling using NMF. It is worth mentioning that moment tensor methods also recover topic models such as LDA, but their assumption is different [Anandkumar et al., 2012c,a,b]. Even though these models are not as fast as the anchor word algorithm, their rich models are worth further consideration in the future.

In the next chapter, we present solutions to address the variability of the datasets by extending the anchor word algorithm using regularization and thoroughly evaluate the anchor word algorithms.

Chapter 3

Regularized Anchor Word Topic Models

3.1 Introduction

The anchor word algorithm for topic models (Section 2.6) is fast and scalable, but our evaluations in this chapter show that it does not produce topics at the high-quality level often observed with Gibbs sampling or variational inference schemes. The anchor word algorithm also lacks the flexibility to address the variability issue in big datasets. To address these issues, we introduce several methods that extend the anchor word algorithm to improve topic quality and to incorporate additional prior knowledge such as informed priors [Jagarlamudi et al., 2012, Zhai et al., 2012]. Our extensions produce new anchor-style algorithms that are more robust and more extensible than the original anchor word algorithm.

Our contribution also includes evaluations of the original anchor word algorithm and the proposed anchor word algorithms using standard topic-modeling metrics such as document held-out likelihood and topic interpretability (for more information about what these metrics are, see Section 2.3). Experimental results show that our proposed anchor word algorithms produce topics that are more interpretable, generalize better when run on held-out data, and can embed external knowledge for richer modeling.

Our enhanced models perform better because they add various useful priors to the anchor word algorithm. Adjusting model priors usually produces better topic modeling [Wallach et al., 2009a, Newman et al., 2011]. We adopt this strategy by interpreting the meaning of priors in a probabilistic sense as statistical regularization and then adding the appropriate regularizers to the objective function of the anchor word algorithm (Equation 2.29). For example, adding an L_2 regularizer is equivalent to assuming that model parameters have a Gaussian distribution. We add an L_2 regularizer to create the L_2-regularized anchor word algorithm.

As described in Section 2.1.2, probabilistic topic models such as LDA use the Dirichlet distribution as a popular prior distribution for both document-topic distributions and topic-word distributions. We formulate the corresponding $Beta$ regularization term for the Dirichlet prior and add that term to the objective function of the anchor word algorithm. We show that the new $Beta$-regularized anchor word algorithm outperforms the standard anchor word algorithm across all evaluation metrics in three datasets: 20 Newsgroups (20NEWS), Neural Information Processing Systems articles (NIPS), and New York Times (NYT) articles. Furthermore, our new model produces high-quality topics for LDA. These new topics are comparable to those produced by Gibbs sampling and variational inference algorithms.

Adding regularization to the anchor word algorithm is a clean solution, because the regularized anchor word models not only retain the sample complexity of the anchor word algorithm but also create more flexibility. In addition, because the additional regularization terms are convex, they do not add significant overhead to the computation time; the new models are still speedy and scalable. With L_2 regularization it is now straightforward to inject external knowledge just by changing the mean of the corresponding Gaussian distribution. This extension increases the variability of the anchor word algorithm. We describe this advantage more deeply in Section 3.6. Finally, due to regularization terms, the new models are less susceptible to noise and are more robust [Bickel et al., 2006, Zou and Hastie, 2005, Tibshirani, 1994, Hastie et al., 2009, Wainwright, 2014].

In the next sections, we lay out the background for this chapter, giving an overview of regularization in optimization and machine learning in Section 3.1.1 and describing the critical role of priors for probabilistic topic models in Section 3.1.2. By connecting these two components, we introduce two regularization terms into the anchor word algorithm to improve its performance and to broaden its practical applications for various datasets.[1]

3.1.1 A Brief Overview of Regularization

In optimization and machine learning, regularization often prevents overfitting and adds robustness to the optimization problem [Girosi et al., 1995, Ng, 2004, Bickel et al., 2006, Zou and Hastie, 2005, Tibshirani, 1994, Hastie et al., 2009, Wainwright, 2014]. This property has become more evident with big-data. With the use of bigger and messier datasets, machine learning applications are being used more extensively [Labrinidis and Jagadish, 2012, Wu et al., 2013, Raghupathi and Raghupathi, 2014]. In those applications, the number of parameters is often much larger than the number of training examples, making most problems ill-posed: they beg for regularizers [Wainwright, 2014].

Additionally, in recent years we have seen a surge of sparse models in many application domains, such as image processing [Mairal et al., 2009, Wright et al., 2009, Majumdar and Ward, 2010, Boureau et al., 2010], natural language processing [Yogatama and Smith,

[1] This chapter revises and extends Nguyen et al. [2014a]. The author's contributions include: deriving mathematical equations, coding and running the experiments, analyzing results, drafting the initial version of the paper, and writing the model and result sections.

2014, Yogatama et al., 2015] and social media [Cha et al., 2015]. Sparse models are desirable in practice, because they are simpler and more interpretable. Most sparse models can be generated using some sparse regularization [Bach et al., 2012, Yuan and Lin, 2006, Jenatton et al., 2011, Baraniuk et al., 2010, Zhao et al., 2009, Huang et al., 2011, Jacob et al., 2009, Morales et al., 2010].

To understand how regularization works, let us start with a very general objective function:

$$c^* = \text{argmin}_c F(c, D),$$

where $c \in R^m$ are model parameters and D are observed variables. In supervised learning, F is the loss function and D is a matrix of observed data and labels [Bishop, 2006, Hastie et al., 2009, Vapnik, 1998]. In unsupervised learning, however, F can have many forms. For example, using *maximum likelihood estimation* (MLE), we form F to maximize the probability that parameters generate the observed data: $F = -\text{log}(P(D \mid c))$.

Overfitting

One of the common issues in machine learning is *overfitting* (Figure 3.1). Overfitting happens when a model has too much freedom to explore the parameter space. For instance, in supervised learning, if the model focuses only on reducing the incurred loss evaluated on the training dataset, it may learn a very peculiar set of parameters c^*. This will result in bad performance on unseen test dataset, and the model will not generalize well. Most MLE models are sensitive to overfitting because more complex models tend to produce a higher likelihood value (small loss) on the training set but a lower value (big loss) on the held-out test set. Regularization is an effective way to address the overfitting problem.

Adding regularization, the above objective function becomes

$$c^* = \text{argmin}_c F(c, D) + \xi \text{Reg}(c),$$

where $\text{Reg}(c)$ is the regularizer that enforces some constraints on model parameters c, and ξ is a regularization weight. The value of ξ controls the effect of $\text{Reg}(c)$ on the final solution of the objective function, balancing the fidelity to the data and the smoothness of model parameters. In practice, ξ is chosen by cross-validation based on how the trained model performs on a development dataset [Goluba et al., 1979, Galatsanos and Katsaggelos, 1992, Lim and Yu, 2016].

Adding a regularization term allows the model to explore only a small region of the parameter space [Hastie et al., 2009], lending the model some unique properties, such as stability against corrupted noise [Wager et al., 2013, Wainwright, 2014]. Regularizers also help to create more scalable and robust algorithms for solving the optimization objective [Wang et al., 2007b, Jerome et al., 2010].

A popular regularization is the L_2 regularizer:

$$\text{Reg}(c)_{L_2} = \|c\|_2 = \sqrt{\sum_i c_i^2}.$$

Figure 3.2, *right* shows how parameters of a linear model ($\text{argmin}\|Ax - b\|$) are learned using the L_2 regularizer. These parameters are smoothly distributed in the solution, since the linear line can touch any point in the circle (or sphere, in high-dimension cases).

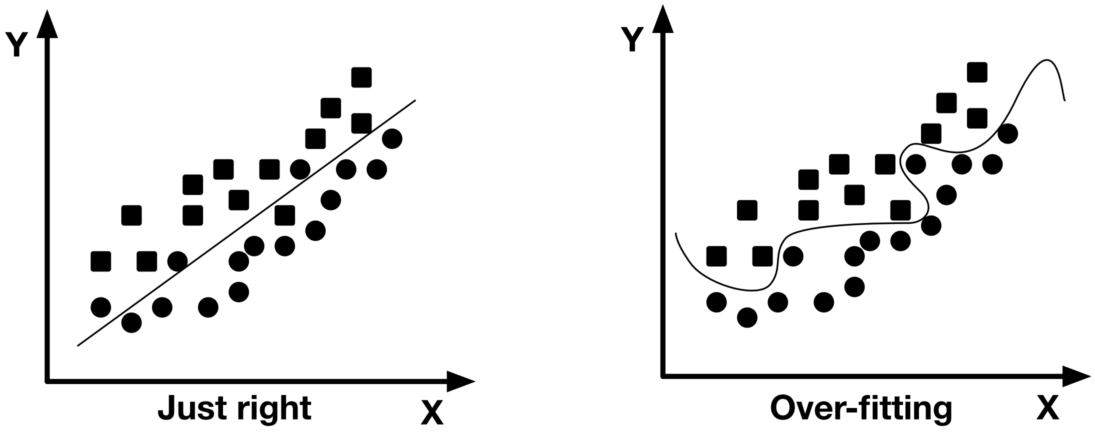


Figure 3.1: A common issue in machine learning: without regularization, the model overfits data.

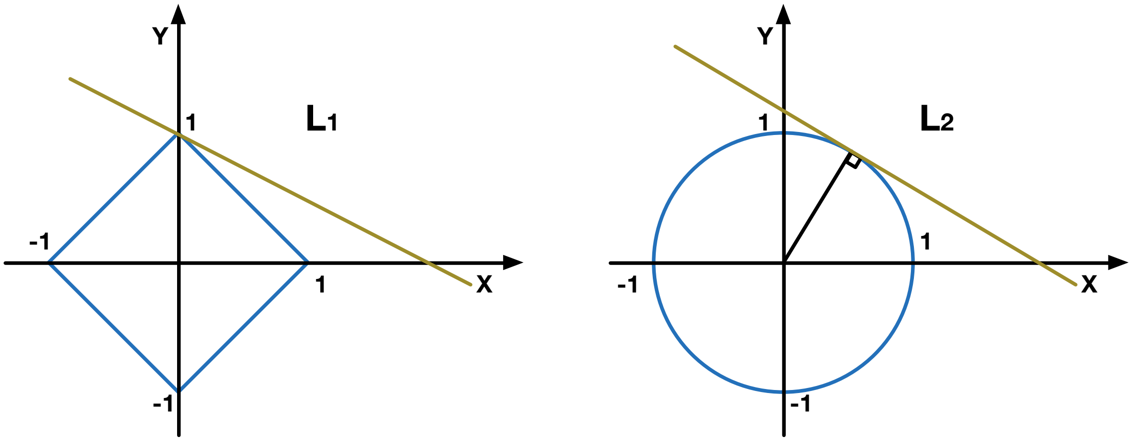


Figure 3.2: L_1 and L_2 regularizers for a linear model. In the case of L_1, the linear line of the solution can touch only one of the vertices and produces a sparse solution, while in the case of L_2, the linear line can move from any direction and touch any point in the circle; it produces a more smooth solution as long as the L_2 norm is equal to one.

Figure 3.1, *right* shows how overfitting occurs without a regularization term; adding L_2 prevents overfitting (Figure 3.1, *left*), making the new model more generalizable on the unseen dataset.

Regularization and Prior Probability

There is a close relationship between the regularization term and the prior distribution assumed for model parameters. For example, Rennie [2003] show that adding an L_2 regularizer is equivalent to assuming model parameters are Gaussian distributed. The L_2 formulation above corresponds to the case of a Gaussian distribution with a zero mean. In practice, we can adjust the mean of the assumed Gaussian distribution if we know a priori about the model parameters. We use this strategy in Section 3.6 to incorporate information from Linguistic Inquiry and Word Count (LIWC) categories [Pennebaker and Francis, 1999] into our L_2-regularized anchor word topic models.

More stable models are not the only requirements; we sometimes need models that are easier to understand and interpret. The most famous example is the case of sparse models. Sparse models are often achieved by using special types of regularization that enforce sparsity [Tibshirani, 1994, Mairal et al., 2014, Bach et al., 2012]. A well-known regularization in this class is the L_1 regularizer that penalizes the absolute sum of parameters,

$$\text{Reg}(c)_{L_1} = \|c\|_1 = \sum_i |c_i|.$$

The model learned using an L_1 regularizer will have zeros for many of its parameters; the model is sparse and interpretable (Figure 3.2, *left*). Eltoft et al. [2006], Kabán [2007] show that an L_1 regularizer is equivalent to a Laplacian prior.

In topic modeling, it is crucial to have sparse and interpretable models, because objects in topic models such as words, topics, and documents are discrete. In LDA, sparsity comes from the Dirichlet distribution priors with parameters less than one. Interestingly, the sparsity property of all anchor word topic models in this thesis comes naturally from solving a nonnegative matrix factorization (see Section 2.5.1). We do not need to use L_1

regularizers to achieve topic sparsity. This property implies that only a small number of K anchor words are used to reconstruct the conditional probabilities of words ($\bar{\boldsymbol{Q}}_i$).

Also, in the topic-modeling community, people pay more attention to priors than to regularizers. The main reason is that most probabilistic models use *posterior inference* to recover topic-model parameters, such as the probability of a topic given a document and the probability of a word given a topic. Posterior inference in a Bayesian sense means that the objective function F is defined as how probable the parameters c are given observation of the data D: $F = -\log(P(c \mid D))$. In machine learning, this formulation has the name of *maximum a posteriori*, or MAP. Using MAP implies that there must be a prior assumption about how the parameters c are distributed. For example, LDA and its related models assume a Dirichlet prior on the document-topic and the topic-word distributions (see Section 2.1.2). We review the role of priors in the next section. We then connect priors back to regularizers to equip the anchor word topic models with the right regularization terms; these regularization terms are shown to be equivalent to popular priors from the Bayesian (probabilistic) perspective.

3.1.2 The Importance of Priors in Topic Modeling

Priors play a crucial role in probabilistic topic models because they appear in all posterior inference problems. Recall that the goal of probabilistic topic models such as LDA is to estimate the posterior distributions of topics given observed words and documents [Blei, 2012]. Given that posterior estimation is intractable, approximation techniques such as Gibbs sampling and variational inference are used to find solutions for topic models (see

Section 2.1.2).

Using approximation techniques underlines the role of priors even more. For example, in Gibbs sampling, we often have to draw samples from prior distributions. So the choice of prior may profoundly affect the convergence of the Gibbs sampling procedure [Griffiths and Steyvers, 2004, Mimno et al., 2008, Resnik and Hardisty, 2010, McCallum, 2002]. Similarly, the conjugacy requirement for variational inference methods depends on prior distributions [Blei et al., 2003, Yee Whye Teh and Welling, 2006, Teh et al., 2006, Wainwright and Jordan, 2008]. Using the right prior distributions speeds up both Gibbs sampling and variational inference schemes for topic models, potentially creating models that are simpler and more scalable.

The choice of priors is critical for topic-model performance as well. Wallach et al. [2009a] show that using an asymmetric Dirichlet prior for the document-topic distribution produces better topic quality and higher document held-out likelihood scores than the usual default symmetric Dirichlet prior used in most LDA implementations. Additionally, they also show that there is not much difference between an asymmetric Dirichlet prior and a symmetric Dirichlet prior in terms of their effect on topic-word distribution. The use of the right priors also increases the robustness of topic models; for example, they are less sensitive to the variation in the number of topics. Probabilistic topic models often use a Dirichlet prior with parameters less than one to create sparsity for the topic-word and document-topic distributions.

Since most probabilistic topic models use Dirichlet prior distributions, we may ask what type of regularization we should use for the anchor word algorithm. We answer this question by analyzing the Dirichlet distribution as the normalization of a vector of

Gamma distributions [Minka, 2000b, Sethuraman, 1994]. We create the $Beta$ regularizer in the form of the negative log likelihood of the appropriate prior distribution. The $Beta$ regularizer is described in detail in Section 3.3.

Informed Priors

The *uninformed* priors that we have just described directly control the general structure of model parameters. However, other types of priors are also possible: priors that model the relationship between model parameters and the data, or between the data themselves. These priors are called informative priors. For example, if we know that specific topics only include certain words, or that some words appear together very often within certain concepts, then using uninformed priors is insufficient to incorporate these constraints. One possibility is to explicitly add constraints to the distribution and control of how words are generated for each topic. This approach works fairly well for LDA, for example, Zhai et al. [2012] build a prior η for each of the LIWC categories so that all words in a category (a topic of interest) have very high prior values compared to other words from the corpus vocabulary. In their work, words such as "afraid", "avoid", or "concern" have very high chances of being assigned to *anxiety* LIWC category. However, this heuristic may not work for the more complex probabilistic models; the computation may also become prohibitive because the inference problem is intractable. We show in Section 3.6 that our method provides a cleaner solution for this problem by changing the mean of the Gaussian (prior) distribution. Experimental results show that the new model successfully incorporates linguistic information from LIWC and produces topics containing correlated words in an expected way.

3.1.3 Chapter Structure

We describe the L_2 regularization for anchor word topic models in Section 3.2. In Section 3.3, we introduce the $Beta$ regularization for modeling the Dirichlet prior in anchor word topic models. We also provide details on how we solve the optimization objective of the anchor word algorithms using the $Beta$ regularizers with an iterative method. We describe three datasets, the NIPS articles, the 20NEWS content, and the NYT articles in Section 3.4. Next, we analyze the performance of our regularized anchor word topic models using two evaluation metrics—document held-out likelihood and topic interpretability—in Section 3.5. Experimental results show that the $Beta$ regularizer creates interpretable topics, and the L_2 regularizer sometimes improves the held-out likelihood. More importantly, we show in Section 3.6 that using L_2 regularization make it easier to incorporate the LIWC category information into the anchor word topic models. We conclude this chapter with a summary and future work in Section 3.7.

3.2 L_2 Anchor: Improving Robustness and Variability

Our primary goal of adding an L_2 regularizer to the anchor word algorithm is to increase the model's robustness, and to provide the flexibility to incorporate external knowledge. Although the choice for the prior distributions of a document over topics and a topic over words in LDA is typically Dirichlet, Dirichlet distributions have been replaced by logistic normals in topic modeling applications [Blei and Lafferty, 2006] and for probabilistic grammars of language [Cohen and Smith, 2009].

In this section, we describe how an L_2 regularization can be used to model topic-word distributions; this is equivalent to assuming that the probability of a word given a topic comes from a Gaussian distribution. Using L_2 regularization will improve model robustness [Wang et al., 2007b, Jerome et al., 2010] and model flexibility.

3.2.1 Objective Function with L_2 Regularization

Recall that in the anchor word algorithm (Section 2.6), each word i is represented as a normalized co-occurrence vector $\bar{\boldsymbol{Q}}_{i,\cdot}$. By the anchor word assumption, each word i will be a linear combination of anchor words $g_k \in \mathcal{G}, \forall k \in [K]$; $\mathcal{G}$ is the set of anchor words. Each combination weight (or coefficient) $C_{i,k}$ corresponds to the probability of observing a topic k given the word i. From there, Bayes' rule (Equation 2.28) will recover the normal topic word distribution as in the standard topic model.

There are many ways we can formulate the objective function for $C_{i,\cdot}$. Two popular approaches are Euclidean distance (square loss) and the Kullback-Leibler divergence (Section 2.6). We use the KL-divergence since it is better for modeling probability distributions. The objective function of the anchor word algorithm using KL-divergence becomes (Equation 2.29)

$$\boldsymbol{C}_{i,\cdot} = \text{argmin}_{C_{i,\cdot}} D_{\text{KL}}\left(\bar{\boldsymbol{Q}}_{i,\cdot} \,||\, \sum_{g_k \in \mathcal{G}} C_{i,k} \bar{\boldsymbol{Q}}_{g_k,\cdot}\right).$$

Given an observed statistics of word-word co-occurrences Q, the above equation uses anchor words vectors $\bar{\boldsymbol{Q}}_{g_k,\cdot}$s to choose the best coefficients that best reconstruct the non-anchor words. We can solve this convex optimization problem (hence ANCHOR objective) for each row $\boldsymbol{C}_{i,\cdot}$ independently, one for each word i in the vocabulary. This property makes the anchor word algorithm very fast because the time complexity only depends on the number of terms. Also, parallelizing the anchor word algorithm is straightforward (see Section 2.6.1). One bottleneck of the anchor word algorithms is that we need to choose good anchor words before performing the recovery step. As discussed in Section 2.6, we do this by using different values of the minimum document frequency threshold M in our experiments. However, it often does not take long to find a good set of anchor words, therefore, compared to Gibbs sampling or variational inference, the anchor word algorithms still run drastically faster.

Since the conditions on the vector $\boldsymbol{C}_{i,\cdot}$ are nonnegative and sum to one, well-established solvers from Lagrangian methods can be used to find the solution. For example, Arora et al. [2013] solve the ANCHOR objective using a straightforward exponentiated gradient algorithm [Kivinen and Warmuth, 1997] with line search [Nocedal and Wright, 2006], and they test convergence by deriving standard KTT systems [Boyd and Vandenberghe, 2014]. Another popular approach is to use L-BFGS gradient optimization [Galassi et al., 2003].

Augmenting the ANCHOR objective function above with an L_2 penalty yields

$$\boldsymbol{C}_{i,\cdot} = \text{argmin}_{C_{i,\cdot}} D_{\text{KL}} \left(\bar{\boldsymbol{Q}}_{i,\cdot} \,||\, \sum_{g_k \in \mathcal{G}} C_{i,k} \bar{\boldsymbol{Q}}_{g_k,\cdot} \right) + \xi \|\boldsymbol{C}_{i,\cdot} - \boldsymbol{\mu}_{i,\cdot}\|_2, \tag{3.1}$$

where regularization weight ξ balances the importance of a high-fidelity reconstruction against the regularization, which encourages the anchor coefficients to be close to the vector $\boldsymbol{\mu}_{i,\cdot}$. When the mean vector $\boldsymbol{\mu}_{i,\cdot}$ is zero, this encourages the topic coefficients to be close to zero. In Section 3.6, we use a non-zero mean $\boldsymbol{\mu}_{i,\cdot}$ to encode an informed prior to encourage topics to discover specific concepts. And using L_2 regularization improves anchor word topic models for a wide range of datasets.[2]

Computationally, adding the L_2 regularizer does not incur any additional resources. The resulting L_2 regularized anchor word algorithm is still fast and parallelizable.

3.3 $Beta$ Anchor: Improving Anchor Topic Quality

As discussed in the Section 2.1.2, two main reasons that make LDA a popular and its topics so good are: (1) LDA is a fully Bayesian model, and (2) LDA uses a Dirichlet prior for both document over topic and topic over word distributions. Using a Dirichlet prior, each document is only represented by a small number of topics, and at the same time, each topic uses only a small number of words. The learning step of LDA involves posterior inference which often results in using inference algorithms such as Gibbs sampling or variational inference. In this section, we want to enforce a Dirichlet prior on the topic over word distribution (the probability of all words given a topic is a V-dimensional multinomial) for the anchor word algorithm. Unlike in the case of the Gaussian prior where we can directly incorporate the corresponding L_2 regularization into the ANCHOR objective, applying a Dirichlet prior is not straightforward. The reason is that the optimization is done on a row-by-row basis in the anchor coefficient matrix C, optimizing C for a fixed word w for all topics. If we want to model the probability of a word, it must be the probability of a word w in a topic versus all other words.

[2] For simplicity, we use the same ξ for all ξ_is. Using different ξ_i for optimizing each $\boldsymbol{C}_{i,\cdot}$ may yield better performance but require huge sacrifice on scalability and speed, we postpone doing this for future work.

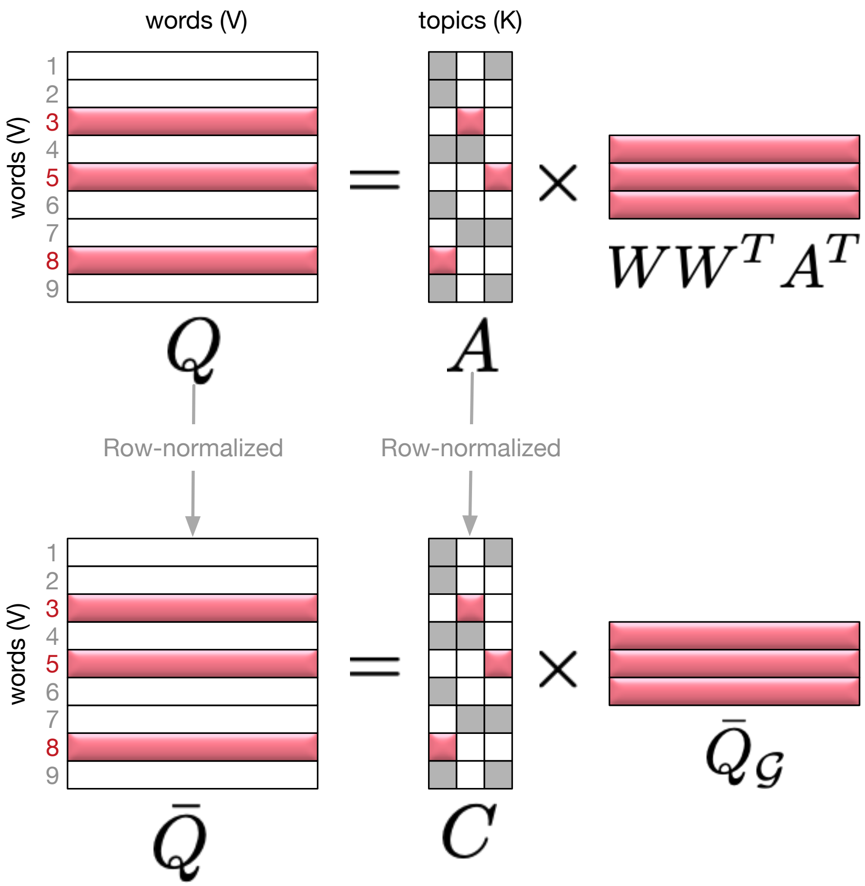


Figure 3.3: Illustration of what anchor words mean. In the matrix A, blank cells mean 0s, other cells (darker color) have values greater than 0. For each column k (topic) of the matrix A, there exist at least one row index i such that $A_{i,k} > 0$ and $A_{i',k} = 0$ if $i \neq i'$. By performing row-normalization on the matrix A to become C, the row in C which corresponds to an anchor word has only one value equal to one while other values are zeros. In this figure, the set of anchor word indexes is: $\mathcal{G} = \{3, 5, 8\}$.

3.3.1 Objective Function with Beta Regularization

Modeling one word versus all other words in a topic is possible. The constructive definition of the Dirichlet distribution [Sethuraman, 1994] states that if one has a V-dimensional multinomial $\theta \sim \text{Dir}(\alpha_1, \ldots, \alpha_V)$, then the marginal distribution of θ_w follows $\theta_w \sim \text{Beta}(\theta_w; \alpha_w, \sum_{i \neq w} \alpha_i)$. This is the tool we need to consider the distribution of a single word's probability.

This requires including the topic matrix A as part of the objective function because the Dirichlet prior is for the columns of A. Recall that $A_{i,k}$ is the probability of the word i in a topic k ($p(w = i \,|\, z = k)$). The topic matrix is a linear transformation of the coefficient matrix C (Equation 2.28):

$$\begin{aligned} A_{i,k} &\propto p(z = k \,|\, w = i) p(w = i) \\ &= C_{i,k} \sum_j \bar{Q}_{i,j}. \end{aligned}$$

The linear transformation of the topic matrix A to become the coefficient matrix C is depicted in the Figure 3.3. The separability property of A carries to C so that we can easily find the solution for the anchor word algorithm.

Because the Dirichlet prior distribution is enforced on the A, the objective for $Beta$ regularization becomes

$$\boldsymbol{C}_{i,\cdot} = \text{argmin}_{C_{i,\cdot}} D_{\text{KL}} \left(\bar{\boldsymbol{Q}}_{i,\cdot} \,||\, \sum_{g_k \in \mathcal{G}} C_{i,k} \bar{\boldsymbol{Q}}_{g_k,\cdot} \right) - \xi \sum_{g_k \in \mathcal{G}} \log(\text{Beta}(A_{i,k}; a, b)), \tag{3.2}$$

where ξ again balances reconstruction against the regularization. To ensure the tractability of this algorithm, we enforce a convex regularization function, which requires that $a > 1$ and $b > 1$. If we enforce a uniform prior—$\mathbb{E}_{\text{Beta}(A_{i,k};a,b)} [A_{i,k}] = \frac{1}{V}$—and that the *mode* of the distribution is also $\frac{1}{V}$, this gives us the following parametric form for a and b:

$$a = \frac{x}{V} + 1, \text{ and } b = \frac{(V-1)x}{V} + 1 \tag{3.3}$$

Algorithm 3 Steps for solving $Beta$ objective function

Input: Co-occurrence matrix $\bar{Q}$ of size $V \times V$,
Set of anchor words $\mathcal{G}$, and a tolerance value ω
Output: Co-efficient matrix $\bar{C}$

```
Initialize $\boldsymbol{C}$ randomly from Dir($\alpha$)
$C_{prev} \leftarrow \mathbf{0}$
$C_{cur} \leftarrow C$
$\Delta C \leftarrow \|C_{cur} - C_{prev}\|_2$
while $\Delta C > \omega$ do
    for $i = 1, 2, \cdots, V$ do
        Solve $C_{i\cdot}$ from Equation 3.2
        using fixed $C_{prev}$ as $C$ in gradient updates
        Subject to: $\sum_k C_{i,k} = 1$ and $C_{i,k} \geq 0$
    $C_{prev} \leftarrow C_{cur}$
    $C_{cur} \leftarrow C$ {update new $C$ from solutions}
    $\Delta C \leftarrow \|C_{cur} - C_{prev}\|_2$
return $C$
```

for real x greater than zero.[3]

3.3.2 Optimizing the Beta Objective Function

To solve the objective with $Beta$ regularization, we will need to compute the gradient function concerning each $C_{i,k}$. Appendix A.3 provides more details on how we compute these gradients. The detailed steps for solving C are described in Algorithm 3. Computationally, adding $Beta$ regularizer is more expensive than adding an L_2 regularizer since we have to loop over the vocabulary several times before the algorithm converges. Practically, this is nothing compared to the case of Gibbs sampling with many expensive iterations until the probabilistic model can find the approximate posteriors.

[3] For $a, b < 1$, the expected value is still the uniform distribution, but the mode lies at the boundaries of the simplex. This case corresponds to a sparse Dirichlet distribution, which our optimization cannot at present model. However, enforcing sparsity for individual element of A is not necessary since the sparsity of the anchor word solution comes from solving the NMF on a whole to reconstruct the observed statistics $\bar{Q}$ (Section 2.5.1).

For solving Equation 3.2, we assume the coefficient matrix C is a constant. After each iteration, we update C based on all updated row values. We initialize C randomly from Dir(α) with $\alpha = \frac{60}{V}$ [Wallach et al., 2009a]. We update C after optimizing all V rows. The newly updated C replaces the old topic coefficients. We track how much the topic coefficients C change between two consecutive iterations i and $i + 1$ and represent it as $\Delta C \equiv \|C^{i+1} - C^{i}\|_2$. We stop optimization when $\Delta C \leq \omega$. When $\omega = 0.1$, the beta regularization typically converges after fewer than ten iterations (Figure 3.9).

3.4 Data Used in Experimentation

We describe three datasets that are used in our experiments. For each dataset we downcase, tokenize, and remove stopwords. Also, we split each dataset into a training set (70%), development set (15%), and a test set (15%): the training data are used to fit models; the development data are used to select parameters (anchor threshold M, document topic prior parameter α, and regularization weight ξ); and final results are reported on the test data. Statistics for the datasets are summarized in Table 3.1.

- Scientific articles from the Neural Information Processing Society: The dataset contains articles from NIPS proceedings 00 to 12 (from the year 1987 to the year 1999).[4] Sam Roweis prepared the dataset from Yann Lecun's raw data using Andrew McCallum's BOW toolkit.

- Internet newsgroups postings: The dataset contains around 18,000 newsgroups posts, categorized into 20 groups (topics) such as hardware, guns, middle east, religions,

[4] `http://cs.nyu.edu/~roweis/data.html`

Corpus	Train	Dev	Test	Vocab
NIPS	1,231	247	262	12,182
20NEWS	11,243	3,760	3,726	81,604
NYT	9,255	2,012	1,959	34,940

Table 3.1: The number of documents in the train, development, and test folds in our three datasets.

and sports.[5] The original dataset is split into a training set (70%) and a test set (30%). We further split the original test set into our dev and test sets equally.

- New York Times articles [Sandhaus, 2008, NYT]: This dataset is the subset of the famous New York Times Annotated Corpus (1.8M documents from January 1987 to June 2007). The NYT dataset contains around 13,000 articles from the New York Times Newsroom.

3.5 Regularization Improving Topic Models

In this section, we measure the performance of our proposed regularized anchor word algorithms. We will refer to specific algorithms in uppercase. For example, the original (unregularized) anchor word algorithm is ANCHOR. Our L_2 regularized variant is ANCHOR-L_2, and our beta regularized variant is ANCHOR-BETA. We compare all anchor word algorithms with two baseline inference algorithms as described in Section 2.2: Gibbs sampling (hence MCMC) and variational inference (hence VARIATIONAL).

We use two evaluation metrics: document held-out likelihood and topic interpretability. As described in the Section 2.3, the held-out likelihood measures how well the model

[5] http://qwone.com/~jason/20Newsgroups/

can reconstruct held-out documents that the model has never seen before. This metric is the standard evaluation for probabilistic models. Topic interpretability is a more recent metric to capture how useful the topics are to human users attempting to make sense of large datasets.

Held-out likelihood cannot be computed with existing anchor word algorithms because the document topic distributions are stochastic and not being recovered. We use the topic distributions learned from the anchor word algorithms as input to a reference variational inference implementation (LDAC package by Blei et al. [2003]) to compute document held-out likelihood. This computation requires an additional parameter, the Dirichlet prior α for the per-document distribution over topics. We select α using grid search on the development set (for different techniques on how to compute held-out likelihood for held-out documents given α, see Wallach et al. [2009b]).

To compute and evaluate topic interpretability produced by topic models, we use the normalized pairwise mutual information (NPMI) over topics' twenty most probable words. Topic interpretability is computed against the NPMI of a reference corpus (see Section 2.3). For topic interpretability evaluations, we use both intrinsic and extrinsic text collections to compute NPMI. Intrinsic topic interpretability is computed on training and development data at development time and training and test data at test time. Extrinsic topic interpretability is computed from English Wikipedia articles, with disjoint halves (1.1 million pages each) for distinct development and testing extrinsic topic interpretability evaluation.

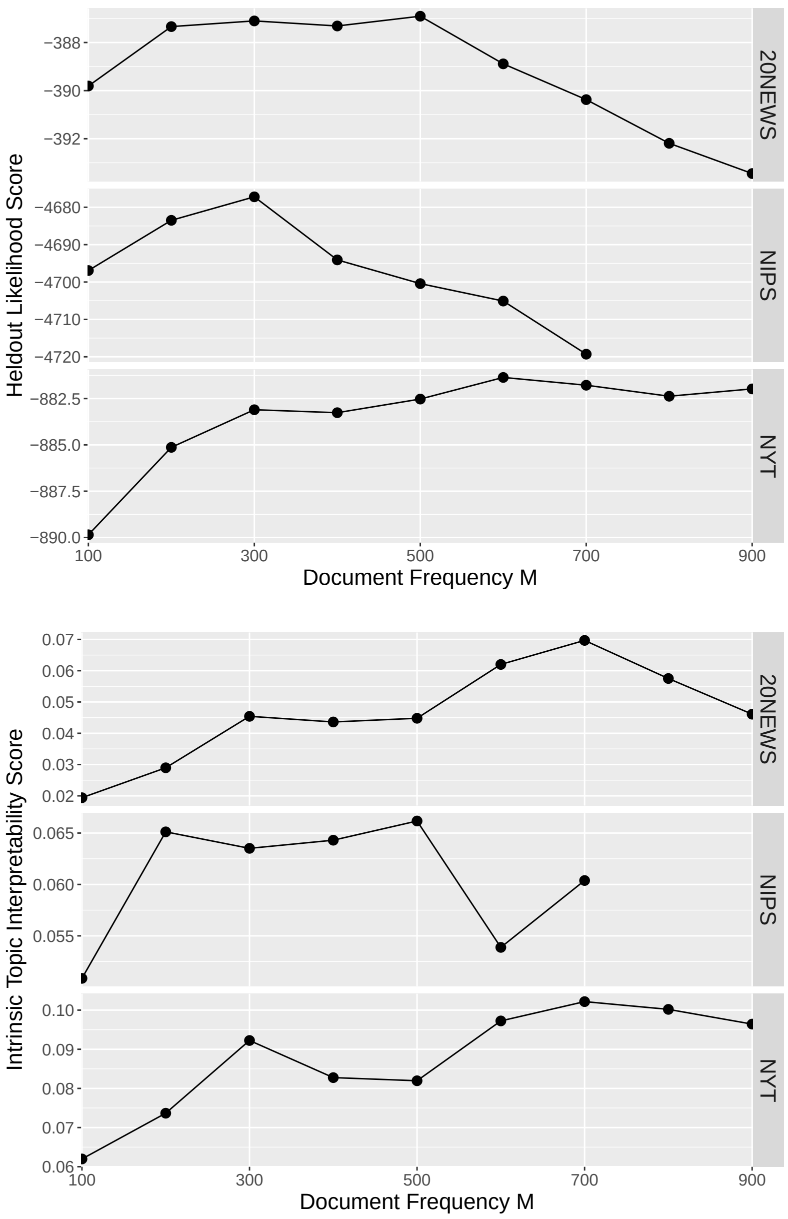


Figure 3.4: Grid search for document frequency M for our datasets with 20 topics (other configurations not shown) on development data. The performance on the held-out likelihood score (top) and the intrinsic topic interpretability score (bottom) indicate that the unregularized anchor word algorithm is very sensitive to M. The M selected here is applied to subsequent models.

3.5.1 Grid Search for Parameters on Development Set

Anchor Threshold A good anchor word must have a unique, specific context but also explain other words well. A word that appears only once will have a very specific co-occurrence pattern but will explain other words' co-occurrence poorly because the observations are so sparse. As discussed in Section 2.6, the anchor word algorithm uses document frequency M as a threshold to only consider words with robust counts.

Because all regularizations benefit equally from higher-quality anchor words, we use cross-validation to select the document frequency cutoff M using the unregularized anchor word algorithm. Figure 3.4 show the performance of the unregularized anchor word algorithm with different M on our three datasets with 20 topics for our two measures document held-out likelihood (*top*) and intrinsic topic interpretability (*bottom*) respectively.

Regularization Weight Once we select a cutoff M for each combination of dataset, number of topics K and an evaluation measure, we select a regularization weight ξ on the development set. Figure 3.6 shows that BETA regularization framework improves intrinsic topic interpretability on all datasets and Figure 3.5 shows the improvement of the held-out likelihood on 20NEWS. Similarly, the L_2 regularization also improves held-out likelihood for the 20NEWS corpus (Figure 3.5).

We do not show the figures for selecting M and ξ using extrinsic topic interpretability, which is similar to intrinsic topic interpretability: ANCHOR-BETA improves extrinsic topic interpretability score on all datasets, ANCHOR-L_2 improves extrinsic topic interpretability score on 20NEWS and NIPS with 20 topics and NYT with 40 topics.

3.5.2 Model Heldout Likelihood

With document frequency M and regularization weight ξ selected from the development set, we compare the performance of those models on the test set. We also compare with

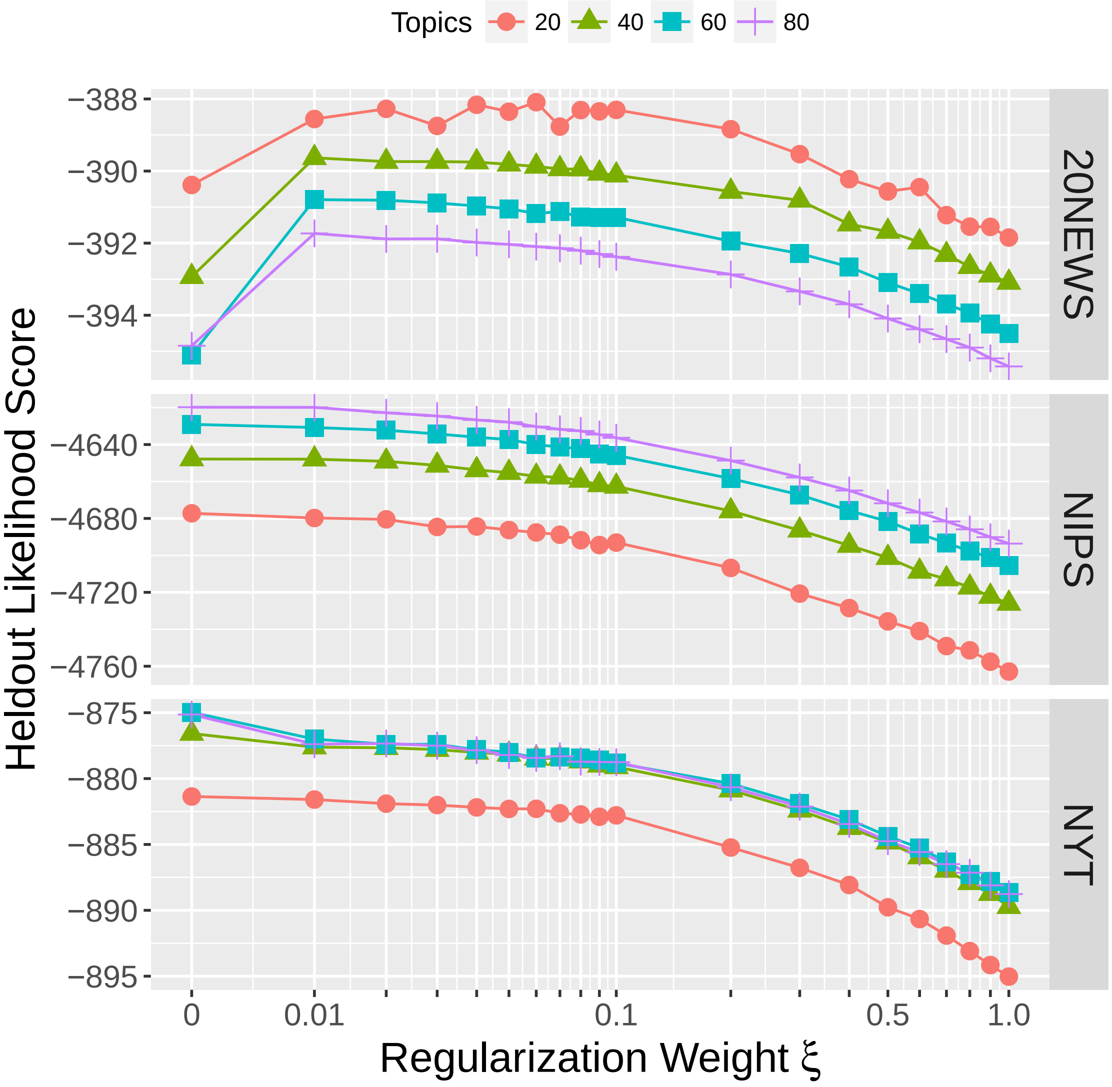


Figure 3.5: Selection of ξ based on heldout likelihood score using ANCHOR-L_2 on the development set. The value of $\xi = 0$ is equivalent to the unregularized anchor word algorithm; regularized versions find better solutions as the regularization weight ξ becomes non-zero. Similar result for ANCHOR-BETA can be found in Section B.1.

standard implementations of LDA: Blei's LDAC (VARIATIONAL) and Mallet (MCMC). We run 100 iterations for LDAC and 5000 iterations for Mallet.

Each result is averaged over three random runs and appears in Figure 3.7. The highly-tuned, widely-used implementations uniformly have better held-out likelihood than anchor word algorithms, but the much faster ANCHOR methods are often comparable.

Within anchor word based methods, ANCHOR-L_2 offers comparable held-out likelihood as unregularized anchor word algorithm.

L_2 **(Sometimes) Improves Generalization** As Figure 3.5 shows, ANCHOR-L_2 sometimes improves held-out development likelihood for the smaller 20NEWS and NIPS corpora. However, the ξ selected on development data does not always improve test set performance. This, in Figure 3.7, ANCHOR-BETA closely tracks ANCHOR. Thus, L_2 regularization does not hurt generalization while imparting expressiveness and robustness to parameter settings.

3.5.3 Topic Interpretability

Beta Improves Interpretability As Figure 3.6 shows, ANCHOR-BETA consistently improves the intrinsic topic interpretability score for the development set, and the ξ selected for development data always improves the test set performance. Figure 3.7 shows that ANCHOR-BETA improves topic interpretability compared to the unregularized anchor word algorithm (ANCHOR). Due to the mismatch between the specialized vocabulary of NIPS and the general-purpose language of Wikipedia, the extrinsic topic interpretability score has a high variance. Therefore, while ANCHOR-BETA often has higher extrinsic interpretability score than ANCHOR, the difference is smaller in the case of the NIPS dataset (the 95% confidence interval of this difference ranges from -0.00593 to 0.0151 and the two-tailed p value equals 0.2594, which is not statistically significant). In the case of larger corpora, such as NYT, $Beta$ regularization helps ANCHOR to learn more interpretable topics (the 95% confidence interval of this difference ranges from 0.00374 to 0.018, and the two-tailed

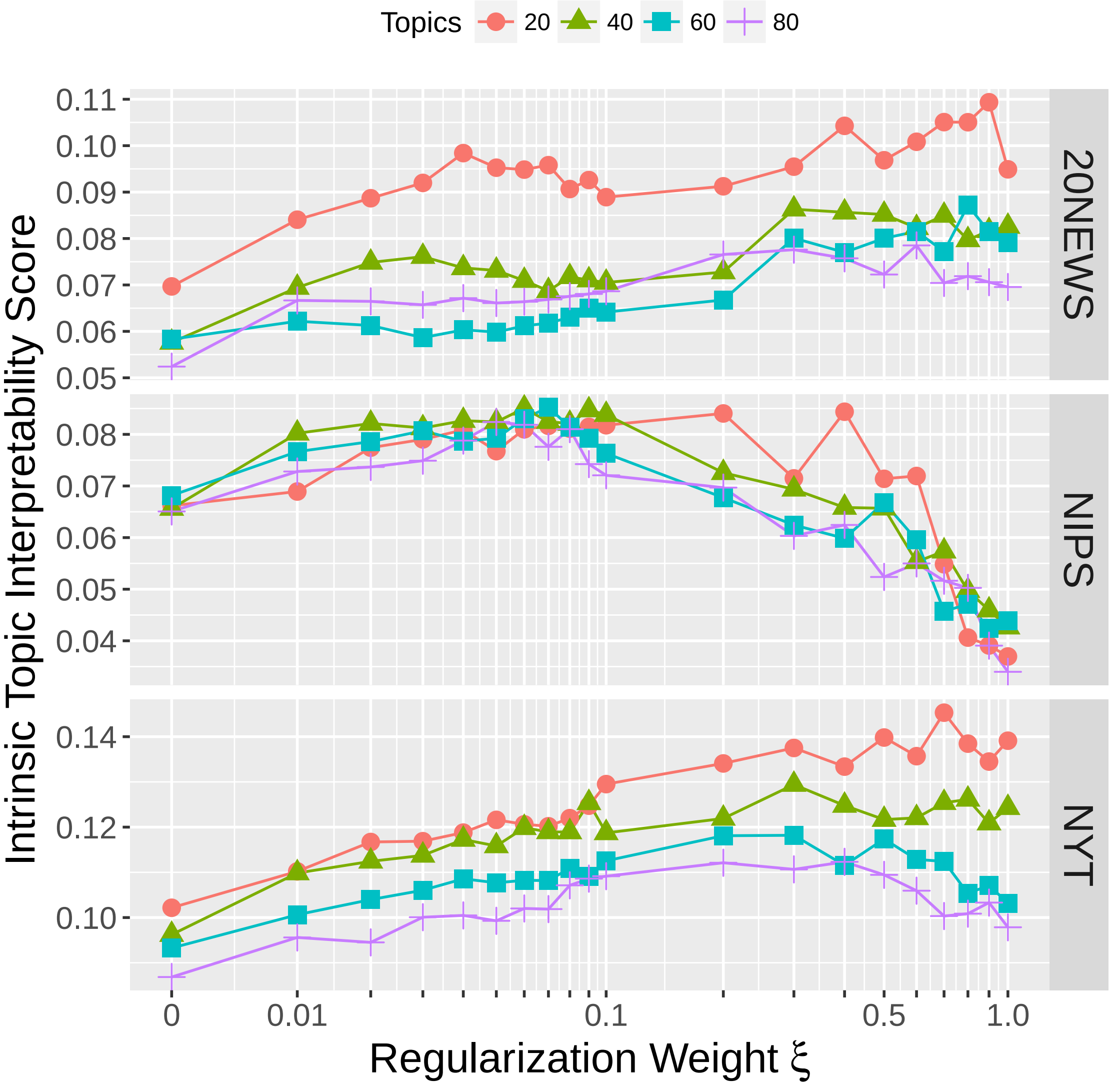


Figure 3.6: Selection of ξ based on intrinsic topic interpretability score using ANCHOR-BETA on the development set. The value of $\xi = 0$ is equivalent to the unregularized anchor word algorithm; regularized versions find better solutions as the regularization weight ξ becomes non-zero. Similar result for ANCHOR-L_2 can be found in Section B.2.

p value equals 0.0167, which is statistically significant). In the following paragraphs, we examine why this is the case.

We first compare the topics from the ANCHOR against ANCHOR-BETA to analyze the topics qualitatively. Table 3.2 shows that $Beta$ regularization promotes rarer words within a topic and demotes common words. For example, in the topic about hockey

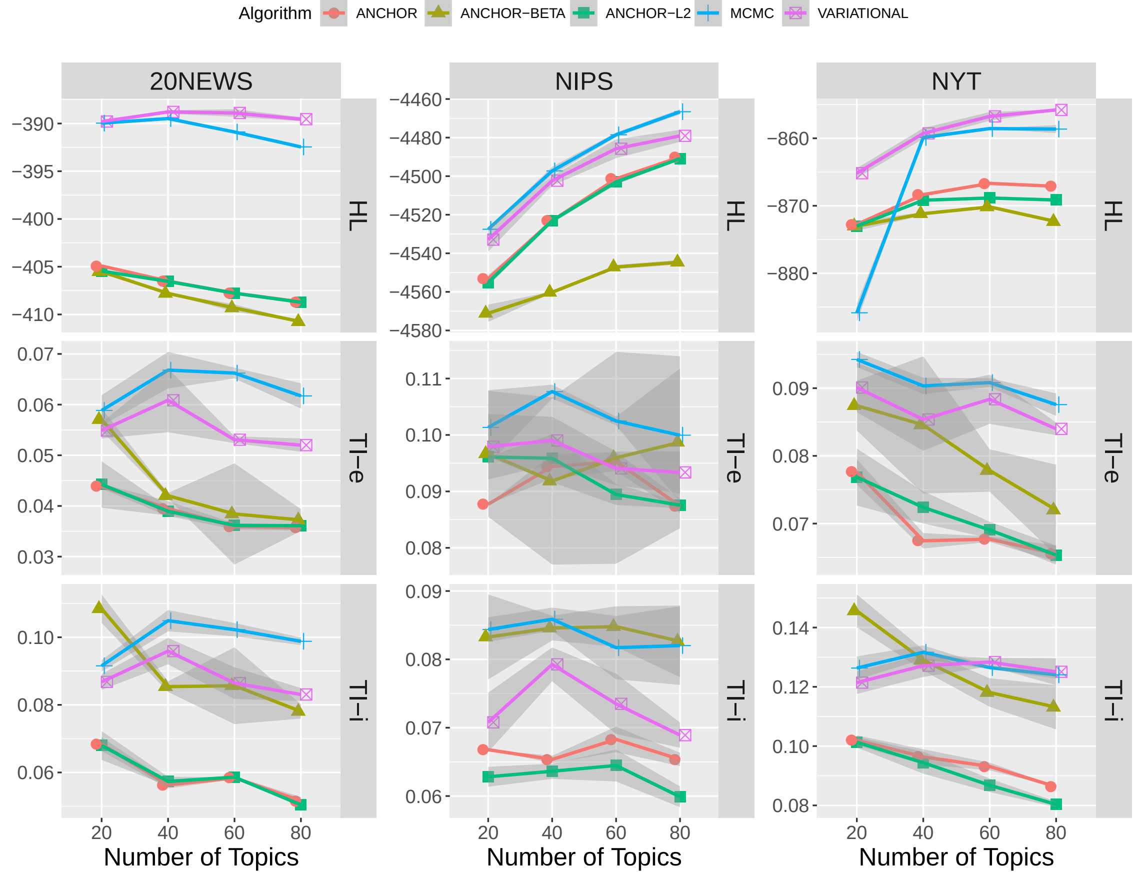


Figure 3.7: Comparing ANCHOR-BETA and ANCHOR-L_2 against the ANCHOR and the traditional VARIATIONAL and MCMC on the held-out likelihood score (HL) and topic interpretability score (TI-e for extrinsic and TI-i for intrinsic). VARIATIONAL and MCMC provide the best held-out generalization. ANCHOR-BETA sometimes gives the best topic interpretability score and consistently produces better topic quality than ANCHOR does. The specialized vocabulary of NIPS causes high variance for the extrinsic interpretability evaluation. In this figure, we use smoothed conditional means with 95% confidence interval to represent data trends.

with the anchor word game, “run” and “good”—ambiguous, polysemous words—in the unregularized topic are replaced by “playoff” and “trade” in the regularized topic. These words are less ambiguous and more likely to make sense to a consumer of topic models.

Figure 3.8 illustrates why this happens. Compared to the unregularized topics from ANCHOR, the *Beta* regularized topics redistribute weights of highly probable terms and create a more uniform distribution. Thus, words that occur highly frequently do not easily

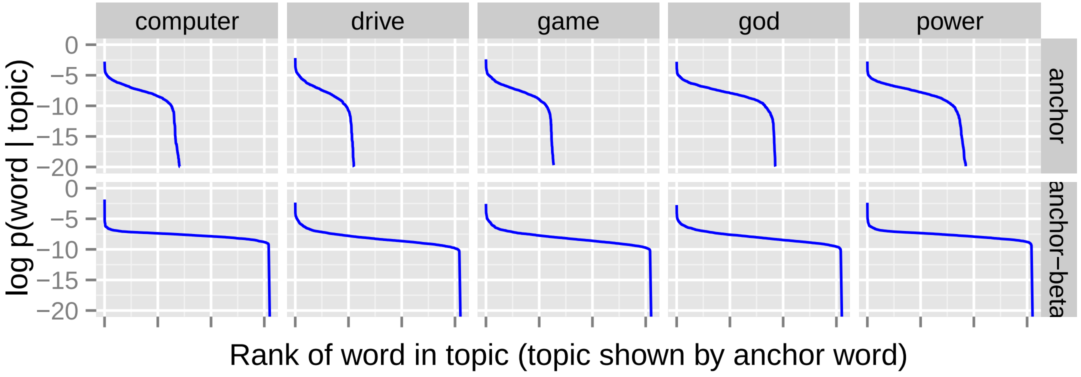


Figure 3.8: How beta regularization influences the topic distribution. Each topic is identified with its associated anchor word. Compared to the unregularized ANCHOR method, ANCHOR-BETA steals probability mass from the "rich" and prefers a smoother distribution of probability mass. These words often tend to be polysemous words that are common across topics.

Topic	Shared Words	ANCHOR (Top, italic) vs. ANCHOR-BETA (Bottom)
computer	computer means science screen	*system phone university problem doe work windows internet software chip mac set fax technology information data* quote mhz pro processor ship remote print devices complex cpu electrical transfer ray engineering serial reduce
power	power play period supply ground light battery engine	*car good make high problem work back turn control current small time* circuit oil wire unit water heat hot ranger input total joe plug
god	god jesus christian bible faith church life christ belief religion hell word lord truth love	*people make things true doe* sin christianity atheist peace heaven
game	game team player play win fan hockey season baseball red wings score division league goal leaf cup toronto	*run good* playoff trade
drive	drive disk hard scsi controller card floppy ide mac bus speed monitor switch apple cable internal port meg	*problem work* ram pin

Table 3.2: A comparison of topics—labeled by their anchor words—from ANCHOR and ANCHOR-BETA. With beta regularization, relevant words are promoted, while more general words are suppressed, thus improving topic coherence.

reach the top of the distribution, and the topics reflect topical, relevant words rather than globally frequent terms.

3.5.4 Discussion

Having demonstrated that regularization can improve the ANCHOR topic modeling algorithm, this section discusses *why* these regularizations can improve the model and the implications for practitioners.

Efficiency Efficiency is a function of the number of iterations and the cost of each iteration. Both ANCHOR and ANCHOR-L_2 require a single iteration, although the iteration of the latter is slightly more expensive. For the BETA regularized anchor word algorithm, as described in Section 3.3, we update the anchor coefficients C row by row and then repeat the process over several iterations until they converge. However, they often converge within ten iterations (Figure 3.9) on all three datasets: this requires many fewer iterations than MCMC or VARIATIONAL inference and the iterations are less expensive. In addition, since we optimize each row $\boldsymbol{C}_{i,\cdot}$ independently, the algorithm can be easily parallelized (see Section 2.6 for more detail).

Sensitivity to Document Frequency While the ANCHOR is sensitive to the document frequency M (Figure 3.4), adding regularization makes this less critical. Both ANCHOR-L_2 and ANCHOR-BETA are less sensitive to M than ANCHOR.

To demonstrate this, we compare the topics of ANCHOR and ANCHOR-BETA when $M = 100$. As Table 3.3 shows, the words “article”, “write”, “don” and “doe” appear in most of ANCHOR’s topics. While ANCHOR-BETA also has some low interpretability topics, it still can find many high interpretability topics, demonstrating ANCHOR-BETA’s greater

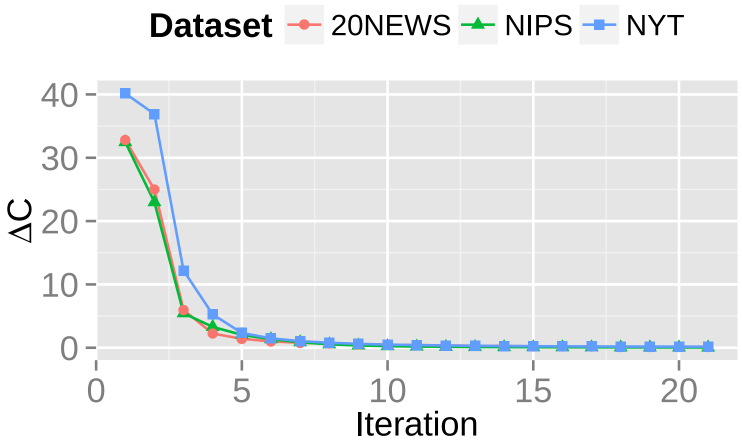


Figure 3.9: Convergence of anchor coefficient C for ANCHOR-BETA. ΔC is the difference between the current C and the C at the previous iteration. C is converged within ten iterations for all three datasets.

Topic	ANCHOR	ANCHOR-BETA
frequently	***article write don doe*** make time people good file question	***article write don doe*** make people time good email file
debate	***write article*** people make ***don doe*** god key government time	people make god ***article write don doe*** key point government
wings	game team ***write*** wings ***article*** win red play hockey year	game team wings win red hockey play season player fan
stats	player team ***write*** game ***article*** stats year good play ***doe***	stats player season league baseball fan team individual playoff nhl
compile	program file ***write*** email ***doe*** windows call problem run ***don***	compile program code file ftp advance package error windows sun

Table 3.3: Topics from ANCHOR and ANCHOR-BETA with $M = 100$ on 20NEWS with 20 topics. Each topic is labeled by its associated anchor word. When $M = 100$, the topics of ANCHOR suffer: the bold and italic words appear in almost every topic. ANCHOR-BETA, in contrast, is less sensitive to suboptimal M.

robustness to suboptimal M.[6]

[6] Some words appear to be the results of tokenization errors.

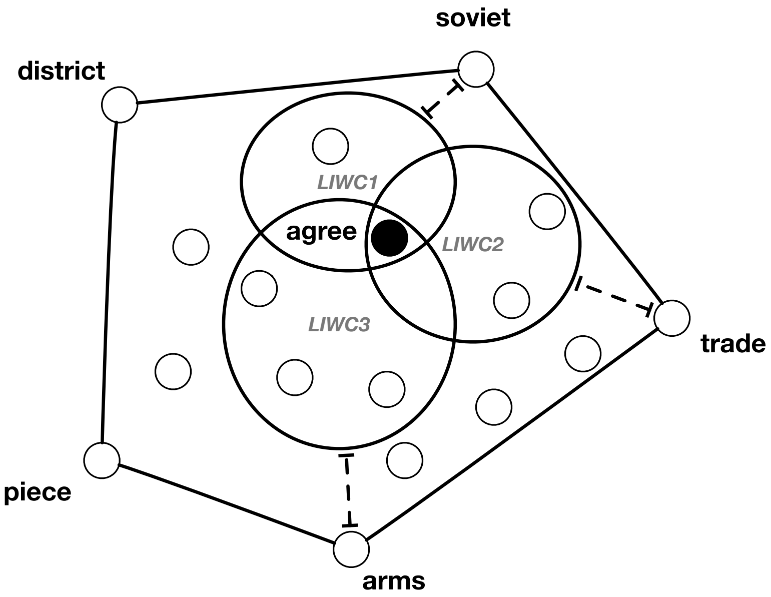
Topic	Shared Words	Original (Top, italic) vs. Informed L_2 (Bottom)
soviet	american make president **soviet** union **war** years	*gorbachev moscow russian force economic world europe political communist lead reform germany country* **military** state **service** washington bush **army** unite **chief troops officer** nuclear time week
district	**assembly** board city **county district member** state york	*representative manhattan brooklyn queens election bronx council island local incumbent housing municipal* **people party group social** republican year make years **friend** vote **compromise** million
peace	american force government israel **peace** political president state unite washington	*war military country minister leaders nation world palestinian israeli election* **offer justice aid deserve** make bush years **fair** clinton **hand**
arms	**arms** bush congress force iraq make north nuclear president state washington weapon	*administration treaty missile defense war military korea reagan* **agree agreement** american **accept** unite **share** clinton years
trade	**administration** america american country **economic** government make president state **trade** unite washington	*world market japan foreign china policy price political* **business economy** congress year years clinton bush **buy**

Table 3.4: Examples of topic comparison between ANCHOR and informed ANCHOR-L_2. Each topic is labeled with its corresponding anchor word. The **bold** words are the informed prior from LIWC. With an informed prior, relevant words appear in the top words of a topic, which also draws in other related terms (underline).

3.6 Injecting Prior Knowledge into Topic Models: Informed Regularization

A common use of priors is to add information to a model (see Section 2.4.3). This is not possible with the current unregularized anchor word method. An informed prior for topic models seeds a topic with words that describe a topic of interest. In topic models, these seeds will serve as a "magnet", attracting similar words to the topic [Zhai et al., 2012].

We can achieve a similar goal with ANCHOR-L_2. Instead of encouraging anchor



$$\mu_{agree,\text{“}soviet\text{”}} = 1/3$$

$$\mu_{agree,\text{“}trade\text{”}} = 1/3$$

$$\mu_{agree,\text{“}arms\text{”}} = 1/3$$

Figure 3.10: Illustration of how to inject LIWC knowledge into anchor word algorithms using L_2 regularization. In this figure, anchor words form a convex hull. The word *agree* belongs to three LIWC categories, which are closest to the three anchor words “soviet”, “trade”, and “arms”. The elements of the prior mean vector $\mu_{agree,\cdot}$ that correspond with these anchor words have values of non-zero (1/3).

coefficients to be zero in Equation 3.1, we can instead encourage word probabilities to be close to an arbitrary mean $\mu_{i,k}$. This vector can reflect expertise in domain knowledge.

As a proof of concept, one example of a source of expert knowledge is Linguistic Inquiry and Word Count [Pennebaker and Francis, 1999, LIWC], a widely used dictionary of keywords related to sixty-eight psychological concepts such as positive emotions, negative emotions, and death. For example, it associates “excessive, estate, money, cheap, expensive, living, profit, live, rich, income, poor, etc.” with the concept of materialism.

We associate each anchor word with its closest LIWC category based on the co-occurrence matrix Q. This is computed by greedily finding the anchor word that has the highest co-occurrence score for any LIWC category. We define the co-occurrence score of a category to an anchor word w_{g_k} as $\sum_i Q_{g_k,i}$, where i ranges over words in this category; we compute the scores of all categories to all anchor words; we then find the highest score and assign the category to that anchor word. We greedily repeat this process until all anchor words have a category.

Given these associations, we create a goal mean $\mu_{i,k}$. If there are L_i anchor words associated with LIWC word i, $\mu_{i,k} = \frac{1}{L_i}$ if this keyword i is associated with anchor word w_{g_k} and zero otherwise. Figure 3.10 illustrates this step for the word *agree*, which appears in three LIWC categories.

We apply ANCHOR-L_2 with informed priors to the NYT dataset with twenty topics and compare these topics against the original topics from ANCHOR. Table 3.4 demonstrates that the topic with anchor word "soviet", when combined with LIWC, draws in the new words "bush" and "nuclear", thus reflecting the threats of force during the cold war. The topic word "arms", when associated with the LIWC category manners with the terms "agree" and "agreement", draws in "clinton", who represented a more conciliatory foreign policy compared to his Republican Party predecessors.

3.7 Conclusions

The anchor word algorithm is a new technique that can analyze large corpora of texts quickly. However, it comes at the cost of the expressive priors common in Bayesian formulations.

This chapter introduces two different regularizations that offer users more interpretable models and the ability to inject prior knowledge without sacrificing the speed and generalizability of the underlying approach. The flexibility of our regularized anchor word models helps us to extend our scope for more applications of topic modeling in a wider range of datasets.

Convex NMF problems, such as the anchor word algorithm, often produce sparse solutions [Ding et al., 2010]. We observe that the anchor word algorithm tends to produce a very sparse solution that promotes general words to the top of the topic distributions, resulting in a low topic interpretability score. We resolve this issue by introducing a smooth Dirichlet prior with $Beta$ regularization. While a $Beta$ regularized anchor topic model produces high interpretability topics, it requires more computation; the ANCHOR-BETA model often runs ten times slower than the original ANCHOR model, though this is still much faster than the non-ANCHOR models.

For ANCHOR-L_2, we observe that more expressiveness sometimes results in a loss of topic interpretability. Depending on the application context, having the expressiveness to incorporate informed knowledge is important, because topics produced by ANCHOR-L_2 are interpretable to domain experts, although these topics may not be interpretable with regard to a general reference corpus, such as Wikipedia. The involvement of humans in evaluating informed topics produced by ANCHOR-L_2 is an exciting future research direction.

A limitation to using regularizations with the anchor word algorithm is the challenge of learning the optimal regularization weights. In our experiments, we use the same regularization weight ξ across the objective functions of all the words; ideally, we should have different regularization weights, with a ξ value for each word i (see Equation 3.1 and Equation 3.2). However, this poses a highly complex combinatoric problem, which cannot be addressed at present.

Incorporating other regularizations into anchor word algorithms could further improve their performance or unlock new applications. Our regularizations function only on a vector of coefficients; applying other regularizations such as structured priors [Andrzejewski et al., 2009b] could efficiently incorporate constraints into topic models.

In the next chapter, we introduce a novel method to incorporate label information into the anchor word algorithms.

Chapter 4

Supervised Anchor Word Topic Models

4.1 Introduction

In this chapter, we continue to address the data variability challenge in the era of big data. Specifically, we focus on large collections of documents in which each document is associated with supervision information such as labels. In addition to the need to extract thematic structures from these document collections, we often have to create statistical models to predict supervision or labels for unseen documents; achieving these two goals concurrently is often very expensive. More challenges come from the size of the datasets; document collections in real-world applications are often huge. Hence, scalability is a requirement for statistical models to be able to deliver real impact. Because the anchor word algorithm is very fast and scalable for extracting themes and topics, we build upon it to introduce a new model called the supervised anchor word algorithm that, while still producing high-quality topics, also makes accurate predictions on unseen documents.

The supervised anchor word algorithm captures the supervision information by combining word-by-word co-occurrence statistics (matrix $\bar{Q}$) with word-by-supervision co-occurrence statistics. This combination enriches the word vector representation in the anchor word algorithm with new dimensions; each dimension corresponds to one supervised piece of information, such as a document label. The experimental results on three sentiment datasets (Amazon product reviews, TripAdvisor hotel reviews, and

Yelp restaurant reviews) demonstrate that the proposed supervised anchor word algorithm produces higher prediction accuracy than both the anchor word algorithm and the SLDA model. In addition, the supervised anchor word algorithm learns specific sentiment anchor words that are indicative of the actual sentiment level that is contained in documents. Because each anchor word corresponds to only one topic, sentiment anchor words capture the sentiment within topics. This property provides a new way in which to gain insights from a large labeled document collection.

Our runtime analysis also indicates that the supervised anchor word algorithm is fast, because it adds no new computational steps to the unsupervised anchor word algorithm. In addition, the supervised anchor word algorithm also inherits the scalability of the unsupervised anchor word algorithm. Therefore, the high-speed and scalable supervised anchor word algorithm is ideal for analyzing large labeled datasets, and it also helps us to analyze information more quickly. To emphasize the advantage of the supervised anchor word algorithm, we compare its runtime with the runtime of supervised topic models in Section 4.6.

Recent probabilistic extensions of LDA that incorporate supervised information tend to add complexity to the model, making the inference process intractable (for details about supervision extensions to LDA, see Section 2.4.2). These supervised topic models often run slowly, limiting the extent to which they can be applied to real-world problems (see Section 4.1.1 below). In contrast, it is straightforward for the supervised anchor word algorithm to use many types of supervision data, such as multiple labels or regression; the representation of words must simply be extended with additional dimensions that reflect the encoded metadata. In the experiment section, we apply this technique only to sentiment datasets (where we have one label per document), but the same strategy can be used for datasets with various types of metadata.

In the subsequent section, we describe several common issues that occur when working with large labeled datasets, and then introduce the supervised anchor topic model as a new tool with which to address them.[1]

4.1.1 Quickly Gaining Insights and Accurately Making Predictions

Every day, the world produces and stores a large amount of data, and a significant portion of this comes with metadata information. For example, most product reviews on the Amazon website feature user ratings, and most Facebook posts are accompanied by some feedback from users such as *emoticons*. Labels of this sort are a common form of metadata, and labels are created to assign certain meanings to data instances. Most labels originate from manual annotation by humans, or from an automatic process performed by applications (e.g., web tagging). It is desirable that analytics tools digest and interpret the meaning from labels in labeled datasets quickly, because their good prediction performance and insights will help companies and organizations—the owners of many huge unlabeled and labeled datasets—make strategic and timely decisions.

Machine learning algorithms have become excellent tools for analyzing large amounts of data, labeled or unlabeled. Unsupervised learning algorithms can now also extract useful patterns from raw datasets. For example, through their ability to produce topics, topic models enable us to better understand very large text corpora. Unsupervised learning

[1] This chapter revises and extends Nguyen et al. [2015a]. The author's contributions include: deriving mathematical equations, coding and running the experiments, analyzing results, drawing figures, and writing most of the paper except the Section 4.6 and the Section 4.5.1 which was written by Jeff Lund.

algorithms, such as the K-means clustering algorithm [Macqueen, 1967, Xu and Wunsch, 2005], can also cluster data instances into groups based on similarities. The main focus of unsupervised learning algorithms is to capture hidden patterns in data through statistical and algorithmic analysis. Training on labeled datasets, supervised machine learning algorithms, such as Support Vector Machines and Decision Trees, can make predictions about new data. These supervised learning algorithms are often called *classifiers*. The ability of classifiers to learn from a historical labeled dataset and then accurately predict labels for new data is essential in many real-world applications. Additionally, understanding customers' buying behaviors also allows us to predict what they are going to buy (Amazon products), the destinations to which they will travel (TripAdvisor reviews), or the restaurants at which they are likely to dine (Yelp reviews).

In many real-world cases, having an accurate prediction is not enough. Take a simple case of marketing products in stores. Using classifiers such as gradient boosted trees, we can provide accurate predictions of customers' propensity to purchase given products. This predictive power is important for marketing current products, but the models provide very little insight into how they make accurate predictions. Without such insight, it is impossible to understand customer behaviors and to design new products. Another example is in designing clinical trials that try to reduce patient burden. Having a good classifier that can accurately predict patient burden is probably not sufficient and is sometimes meaningless; what is more important is to understand the relationships among many factors (features) that may affect how patients feel the burden in clinical trials. These examples highlight the importance of having both predictive performance and insight to help with decision-making strategies.

To address the challenge, machine learning researchers have combined the power of unsupervised models to generate insights and the power of supervised models to make accurate predictions. One important combination is of topic modeling algorithms and supervised learning algorithms, which is termed supervised topic modeling. Supervised topic models are algorithms that capture insights and produce accurate predictions for large labeled corpora; insights come from topic modeling, and accurate predictions come from supervised learning. One challenge that supervised topic models face is the scalability and the sluggish performance on large corpora because most models rely on slow probabilistic inference schemes. For a more thorough review of different supervised topic models and their performance, see Section 2.4. It is worth noting that this idea is different from semi-supervised learning algorithms where the main goal is solely prediction; semi-supervised learning algorithms address the issue of a limited number of labeled data points for prediction by using a large number of unlabeled data points to create better classifiers than those that use labeled data alone [Chapelle et al., 2009, Zhu et al., 2009b, Turian et al., 2010, Erhan et al., 2010, Kingma et al., 2014].

Trying to improve the scalability and speed of unsupervised and supervised learning algorithms for large datasets is not new. Many approaches focus on using scalable and parallel infrastructures, such as powerful hardware devices, and networking and communication architectures for transferring and processing large datasets. Common examples include big data packages, such as Map Reduce and Spark, and GPU-based computation. Other directions focus on scaling up various classifiers to handle very large labeled datasets while maintaining high prediction accuracy. Algorithmic progress has also played a central role; for example, slow and high-quality discriminative algorithms, such as Support Vector Machines (SVMs), presently run very quickly due to advances in online learning (e.g., the Vowpal Wabbit package) [Bottou, 1998, Bottou and Cun, 2003, Langford et al., 2009, Duchi et al., 2010]. Online learning also helps with applications in matrix factorization [Mairal et al., 2010] or topic modeling [Hoffman et al., 2010, Zhai and Boyd-Graber, 2013]. Furthermore, many research projects in machine learning have focused on improving the scalability of statistical unsupervised learning algorithms. This direction has taken many forms, such as the creation of online approximations of large batch algorithms [Hoffman et al., 2013, Zhai et al., 2014] or improvements in the efficiency of sampling [Yao et al., 2009b, Hu and Boyd-Graber, 2012, Li et al., 2014].

These insights have also improved supervised topic models. For example, Zhu et al. [2013] formulate the MedLDA max-margin supervised topic models [Zhu et al., 2009a] so that the hinge loss is included inside a collapsed Gibbs sampler rather than applied externally on the sampler using costly SVMs. Using insights obtained from Smola and Narayanamurthy [2010], the authors improve the samplers to run in parallel to train the model. While these advancements have enhanced the scalability of max-margin supervised topic models, the improvement is limited by the fact that the sampling algorithm grows with the number of tokens. Another direction explores the idea of using efficient representations of summary statistics to estimate statistical models. We have witnessed the success of this approach for unsupervised models [Cohen and Collins, 2014]. For supervised models, Wang and Zhu [2014] demonstrate how to use tensor decomposition instead of sampling to estimate the parameters of SLDA to find maximum likelihood estimates.

In contrast, our supervised anchor word algorithm builds upon a very fast and scalable unsupervised anchor word topic model and enriches its representation to capture metadata and label information to help with prediction while still providing insight from topics. Our supervised anchor word algorithm predicts very accurately in three sentiment datasets and provides particular anchor words for topics, which can be used to explain the meaning of labels. One way in which it is distinguished from most probabilistic supervised topic models is that the supervised anchor word algorithm focuses on generating predictive topic features for supervised classifiers (such as SVM or logistic regression) rather than on directly making predictions itself. The advantage of this approach, as seen in a later section, is that the topics produced by the supervised anchor word algorithm are very insightful about sentiments and can be used to explain why they are accurate for predictions.

4.1.2 Chapter Structure

We introduce the supervised anchor word algorithm in Section 4.2 then, in Section 4.3, describe three sentiment datasets used in our experiments: Amazon product reviews, TripAdvisor hotel reviews, and Yelp restaurant reviews. The approach we introduce is general, but we use sentiment as the focus to illustrate and validate the model. We present the main quantitative results in Section 4.4, in which we compare the prediction performance of the proposed supervised anchor word algorithm with that of the original anchor word algorithm, LDA, and SLDA. Section 4.5 analyzes the benefit of sentiment anchor words for prediction and for extracting insights from sentiment documents. Finally, we compare the runtimes of different topic models in Section 4.6, and summarize the current chapter in Section 4.8.

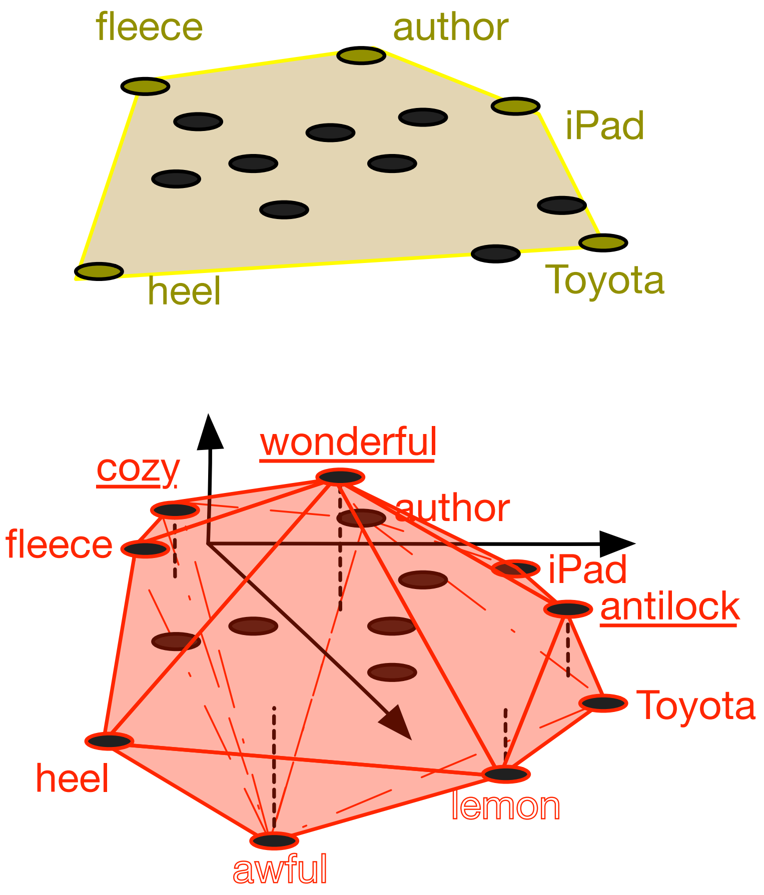


Figure 4.1: Geometric intuition behind supervised anchor words. Anchor words form the convex hull of word co-occurrence probabilities in unsupervised anchor topic modeling (top). Adding an additional dimension to capture metadata, such as sentiment, changes the convex hull: positive words appear above the original 2D plane (underlined) and negative words appear below (in outline).

4.2 SUPANCHOR: Incorporating Supervision into the Co-occurrence Matrix

Our goal is create a fast and scalable supervised topic model that can make accurate prediction on labeled datasets. Because the anchor word algorithm scales so well compared to traditional probabilistic inference, we now unify the supervised topic models of Section 2.4.2 with the anchor word algorithm discussed in Section 2.6. We do so by aug-

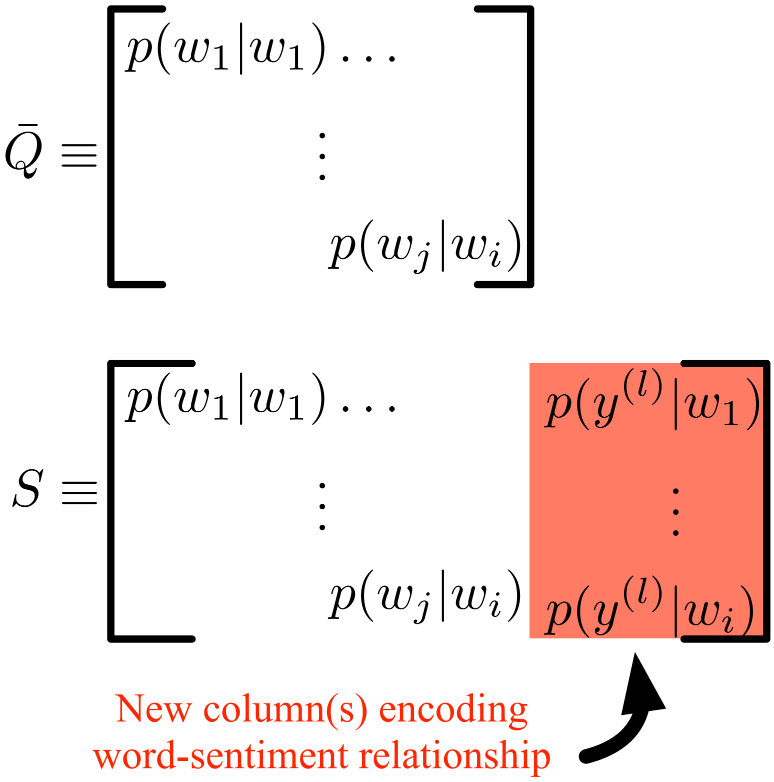


Figure 4.2: We form a new column to capture the relationship between words *and* each sentiment level: per entry is the conditional probability of observing a sentiment level $y^{(l)}$ given an observation of the word w_i. Adding all of these columns to $\bar{Q}$ forms an augmented matrix S.

menting the word co-occurrence matrix $\bar{Q}$ with an additional dimension for each metadata attribute, such as sentiment. We provide the geometric intuition in Figure 4.1. Unlike models in Section 2.4.2, our approach does not try to predict labels directly but learns good representations of words and topics that capture label information from training set; prediction of labels for unseen documents is still performed using traditional supervised learning algorithms such as SVM (see below).

Picture the anchor words projected down to two dimensions [Lee and Mimno, 2014a]: each word is a point, and the anchor words are the vertices of a polygon encompassing every point. Every non-anchor word can be approximated by a convex combination of the anchor words (Figure 4.1, top).

Now add an additional dimension as a column to $\bar{Q}$ (Figure 4.2). This column encodes the metadata specific to a word. For example, we encode sentiment metadata in a new dimension (Figure 4.1, bottom). Neutral sentiment words will stay in the plane inhabited by the other words, positive sentiment words will move up, and negative sentiment words will move down. For simplicity, we only show a single additional dimension, but in general, we can add as many dimensions as needed to encode the metadata.

In this new space, some of the original anchor words may still be anchor words ("author"); these are sentiment-neutral topic oriented words. Other words that were near the convex hull boundary in the unaugmented representation may become anchor words in the augmented representation because they capture both topic and sentiment ("anti-lock" vs. "lemon")—"anti-lock" is a topic-specific sentiment bearing word in the context of automobiles. Finally, some words might become anchor words in the new higher-dimensional space because they are important for explaining extreme sentiment values ("wonderful" vs. "awful").

4.2.1 Enriching Word Vector Representations with Supervision

Having explained how a word is connected to sentiment, we now elaborate on how to model that connection using the metadata element such as conditional probability of sentiment given a particular word. Assume that sentiment is discretized into a finite set of L sentiment levels $\{y^{(1)}, y^{(2)}, \ldots, y^{(L)}\}$ and that each document is assigned to one of these levels. We define a matrix S of size $V \times (V + L)$. The first V columns are the same as $\bar{Q}$ and the L additional columns capture the relationship of a word to each discrete sentiment level.

For each additional column l, $S_{i,(V+l)} \equiv p(y = y^{(l)} \,|\, w = i)$ is the conditional probability of observing a sentiment level $y^{(l)}$ given an observation of word i. We compute the conditional probability of a sentiment level $y^{(l)}$ given word i

$$S_{i,(V+l)} \equiv \frac{\sum_d (\mathbb{1}\left[i \in d\right] \cdot \mathbb{1}\left[y_d = y^{(l)}\right])}{\sum_d \mathbb{1}\left[i \in d\right]}, \tag{4.1}$$

where the numerator is the number of documents that contain word type i and have sentiment level $y^{(l)}$ and the denominator is the number of documents containing word i.

Given this augmented matrix, we again want to find the set of anchor words $\mathcal{G}$ and coefficients $C_{i,k}$ that best capture the relationship between words and sentiment (cf. Equation 2.27)

$$S_{i,\cdot} = \sum_{g_k \in \mathcal{G}} C_{i,k} S_{g_k,\cdot}. \tag{4.2}$$

Because we retain the property that non-anchor words are explained through a linear combination of the anchor words, our method retains the same theoretical guarantees of sampling complexity and robustness as the anchor word algorithm (Section A.4 detailes analysis on this property).

To facilitate direct comparisons between the anchor word algorithm and the supervised anchor word algorithm, we keep the number of anchor words fixed in our experiments. Even so, the introduction of metadata forces the anchor method to select the words that best capture this metadata-augmented view of the data. Consequently, some of the original

Corpus	Train	Test	Vocab	Class **+1**
AMAZON	13,300	3,314	2,662	52.2%
TRIPADVISOR	115,384	28,828	4,867	41.5%
YELP	13,955	3,482	2,585	27.7%

Table 4.1: Statistics of the datasets in our experiments.

anchor words will remain, and some will be replaced by sentiment-specific anchor words.

4.3 Sentiment Datasets

We use three common sentiment datasets for evaluation: Amazon product reviews [Jindal and Liu, 2008, AMAZON], Tripadvisor hotel reviews [Wang et al., 2010, TRIPADVISOR], and Yelp restaurant reviews [Jo and Oh, 2011, YELP]. For each dataset, we preprocess by lowering case, tokenizing, and removing all non-alphanumeric words and stopwords. As concise reviews are often inscrutable and lack cues to connect to the sentiment, we only consider documents with at least thirty words. We also reduce the vocabulary size by keeping only words that appear in a sufficient number of documents: fifty for AMAZON and YELP datasets, and one hundred and fifty for TRIPADVISOR.[2]

Our goal is to perform binary classification of sentiment. Due to a positive skew of the datasets, the median for all datasets is four out of five. It is not surprising that in most 4-star reviews, we often see negative comments on features of products (e.g., in Amazon reviews), hotel amenities (e.g., in TripAdvisor reviews), or restaurant issues (e.g., in Yelp reviews); the language used in most 4-star reviews are positive however there is always a 'but' to reflect some unsatisfactory features. Hence, we split the at the median: all 5-star

[2] Note that this preprocessing may remove important signal, for example, "I hate my Subaru" vs "I HATE my Subaru!!!!".

reviews are assigned to class **+1** and the rest of the reviews are assigned to class **-1**.

Furthermore, we divide each sentiment dataset into five random folds for 5-fold cross-validation. We used four folds to form the TRAIN set and reserved the last fold for the TEST set. Table 4.1 summarizes the composition of each dataset and the percentage of documents with high positive sentiment. Details about each sentiment dataset are in the following:

- Amazon Product Reviews: This is a subset that contains five product categories such as computers, MP3 players, and GPS devices from the Amazon Product Reviews (about 5.8 million reviews) crawled from amazon.com in June 2006 [Jindal and Liu, 2008, Nguyen et al., 2014b]. This subset contains roughly 37,000 reviews of the top fifty products with the most reviews. Each review is associated with one sentiment value from one to five. After preprocessing, we have around 17,000 reviews with sufficient length.

- Tripadvisor Travel Reviews: Authors from Wang et al. [2010] crawled about 236,000 hotel reviews from tripadvisor.com from February 14, 2009 to March 15, 2009. This dataset also has ratings from one to five. After preprocessing, we have around 144,000 reviews.

- Yelp Restaurant Reviews: This dataset was collected by Jo and Oh [2011]. It contains 30,000 reviews of the 320 most rated restaurants in four cities: Atlanta, Chicago, Los Angeles, and New York City. After preprocessing, we are left with around 17,500 long reviews.

4.4 Sentiment Prediction using Supervised Topic Models

In this section, we evaluate the effectiveness of our new method on a binary sentiment classification problem. Specifically, we compare the supervised anchor word algorithm and the original unsupervised anchor word algorithm for classification regarding both accuracy and speed.

4.4.1 Documents to Labels

We compare the effectiveness of different representations in predicting high-sentiment documents: unsupervised topic models (LDA), traditional supervised topic models (SLDA), the anchor word algorithm (ANCHOR), our supervised anchor word algorithm (SUPANCHOR), and a traditional tf-idf [Salton, 1968, TF-IDF] representation of the words.

The anchor word algorithms only provide the topic distribution over words; they do not provide the per-document assignment of topics needed to represent the document in a low-dimensional space as necessary for producing a prediction y_d. Fortunately, this requires only a very quick—because the topics are fixed—pass over the documents using a traditional topic model inference algorithm. We use the variational inference implementation for LDA of Blei et al. [2003] to obtain $\bar{z}_d$, the topic distribution for document d. After estimating topic proportions on the TRAIN, we directly use native variational inference of LDAC to apply pre-trained topics to extract DEV and TEST topic proportions (see Section 3.5).

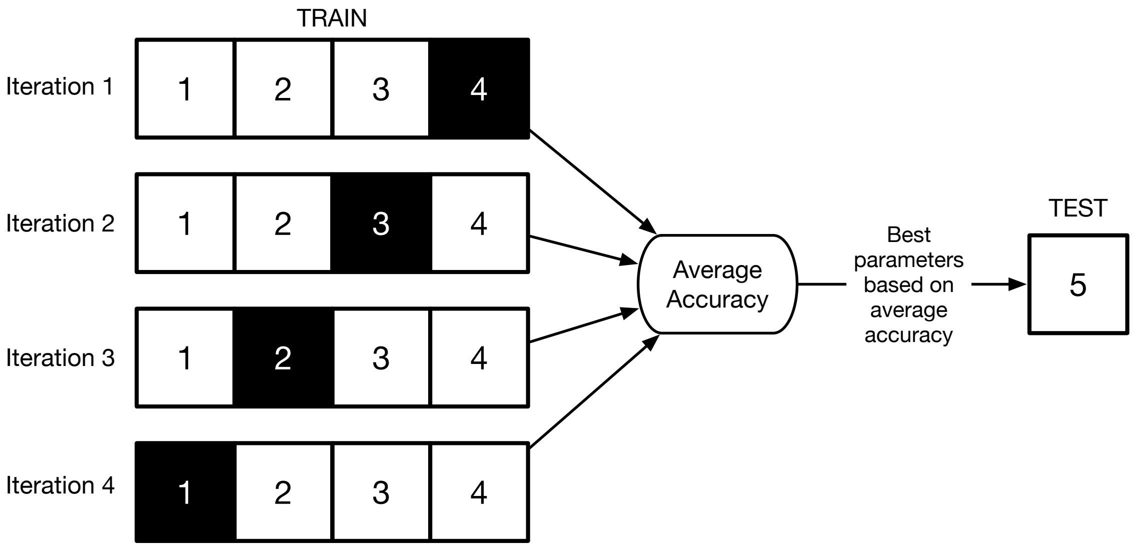


Figure 4.3: We split each dataset into five folds, fold-5 is reserved for the TEST set and use the first four folds (TRAIN) to performance 4-fold cross-validation to find the best set of parameters. In cross-validation, the DEV set is rotated.

Classifiers Given a low-dimensional representation of a test document, we predict the document's sentiment y_d. We have already inferred the topic distribution $\bar{z}_d$ for each document, and we use $log(\bar{z}_d)$ as the features for a classifier. Feature vectors from training data are used to train the classifiers, and feature vectors from the development or test set are used to evaluate the classifiers.

We run three standard machine learning classifiers: decision trees [Quinlan, 1986], logistic regression [Friedman et al., 1998], and a discriminative classifier. For decision trees (hence TREE) and logistic regression (hence LOGISTIC), we use SKLEARN.[3] For the discriminative classifier, we use a linear classifier with hinge loss (hence HINGE) in Vowpal Wabbit.[4] Because HINGE outputs a regression value in $[0, 1]$, we use a threshold 0.5 to make predictions.

[3] http://scikit-learn.org/stable/

[4] http://hunch.net/~vw/

Parameter Tuning Parameter tuning is important in topic models, so we cross-validate. As mentioned earlier, each sentiment dataset is split randomly into five folds, four folds for the TRAIN set and one fold for the TEST set. All cross-validation results are averaged over the four held out DEV sets (hence this is four-fold cross-validation); the best cross-validation result provides the parameter settings we use on the TEST set (Figure 4.3).

For ANCHOR and SUPANCHOR, the parameter for the document-level Dirichlet prior α is required for inferring document-topic distributions given learned topics. Despite selecting this parameter using grid search, α does not affect our final results. The same is also true for SLDA: its predictive performance does not significantly vary as α varies, given a fixed number of topics K. We use the SLDA implementation by Chong Wang to estimate α.[5]

Anchor word algorithms are sensitive to the value of anchor threshold M (the minimum document frequency for a word to be considered an anchor word). For each number of topics K, grid search finds the best value of M. Figure 4.4 shows the performance trends.

For LDA, we use the Gibbs sampling implementation in Mallet.[6] For training the model, we run LDA with 5,000 iterations, with a lag of 5 and a 50 iteration burn-in period; and for inference (on DEV and TEST) of document topic distribution we iterate 100 times, with a lag of 5 and a 50 iteration burn-in period. As Mallet accepts $\sum \alpha_i$ as a parameter, we always initialize $\sum \alpha_i = 1$ and only grid search over different values of β, the hyperparameter for Dirichlet prior over the per-topic topic-word distribution, starting from 0.01

[5] http://www.cs.cmu.edu/~chongw/slda/

[6] http://mallet.cs.umass.edu/topics.php

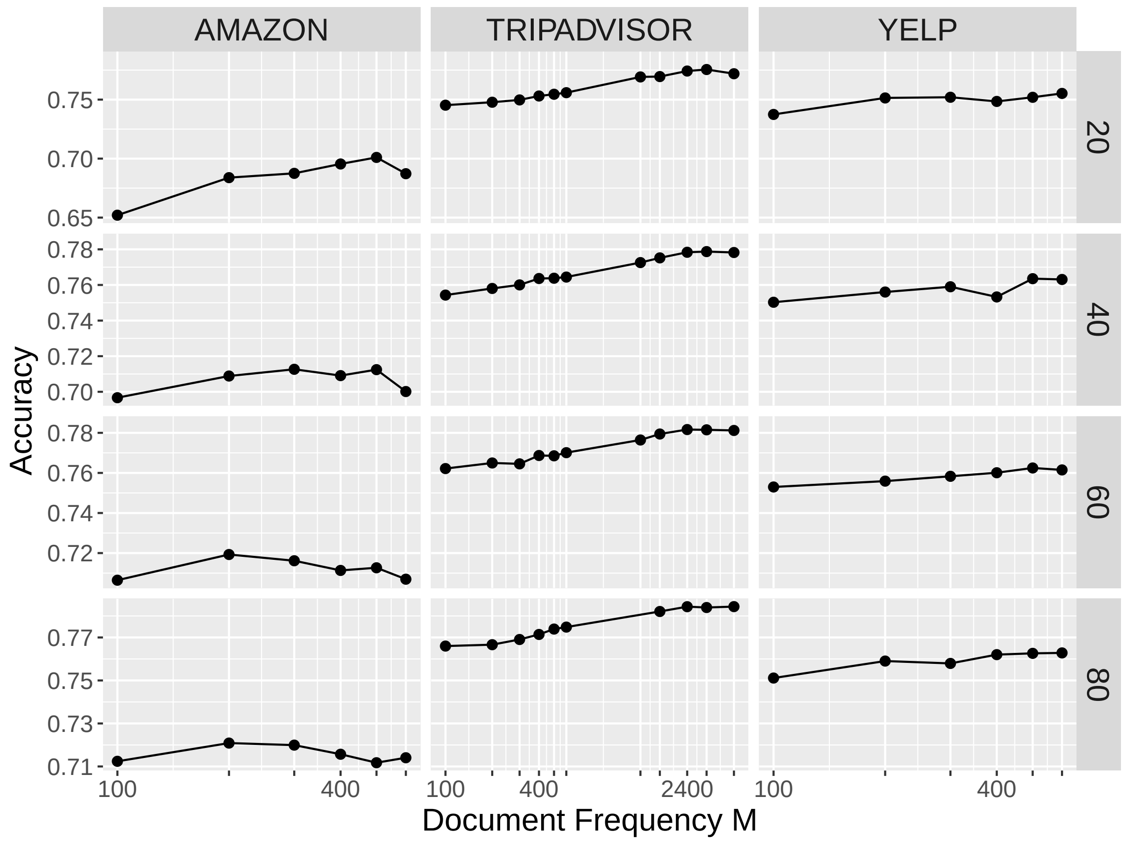


Figure 4.4: Grid search for selecting the anchor word document threshold M for SUPANCHOR based on development set accuracy.

and doubling until reaching 0.5.

4.4.2 Topic Features

This section evaluates how topics produced by topic models predict sentiment. Learning topics that jointly reflect words and metadata improves subsequent prediction. The results for both SUPANCHOR and ANCHOR on the TEST set are shown in Figure 4.5. SUPANCHOR outperforms ANCHOR on all datasets. This trend holds consistently for the LOGISTIC, TREE, and HINGE methods for sentiment prediction. For example, with twenty topics on the AMAZON dataset, SUPANCHOR gives an accuracy of 0.71 in comparison to only 0.62

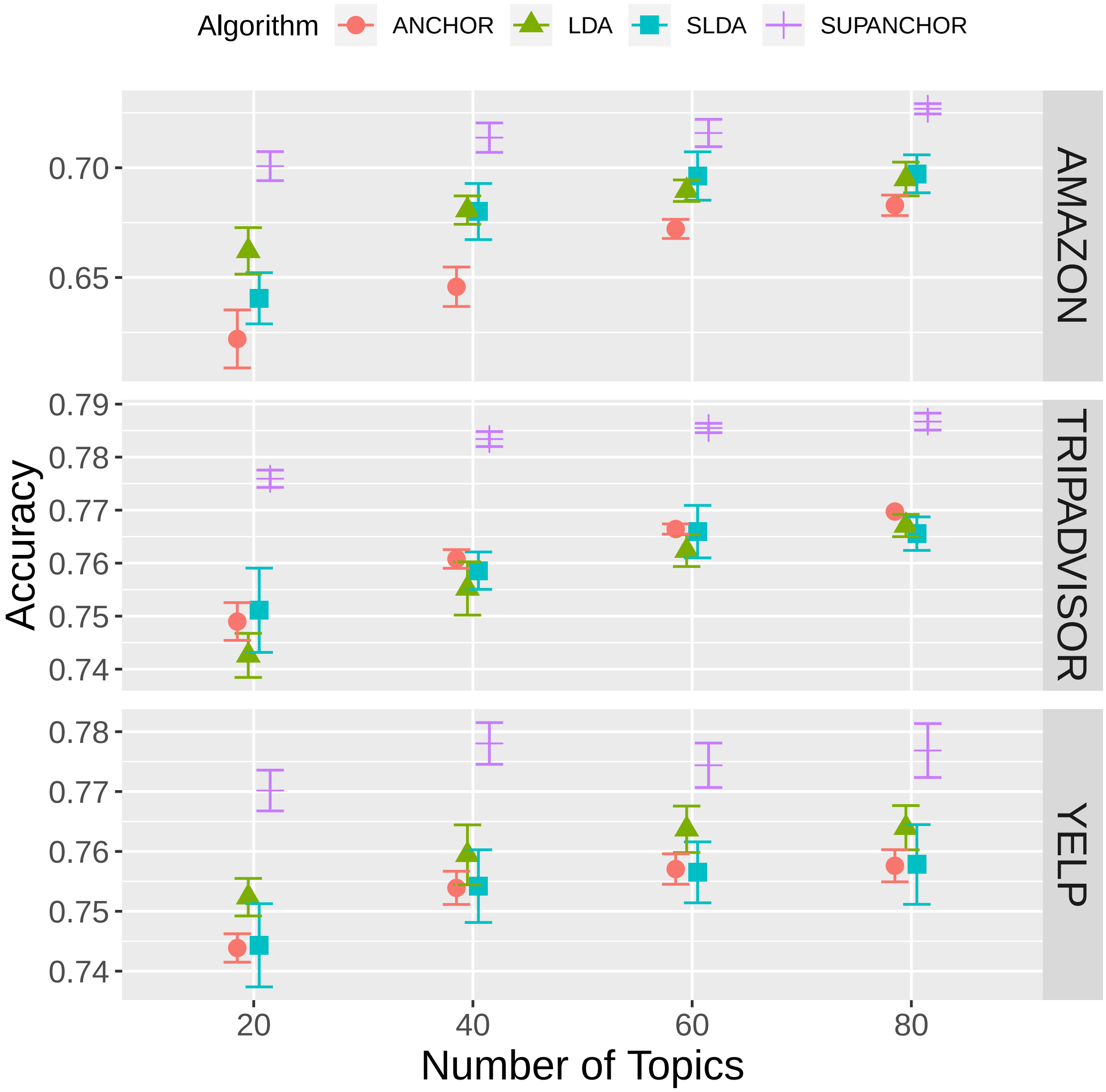


Figure 4.5: Mean accuracy on TEST fold. Error bars indicate 95% confidence intervals. SUPANCHOR outperforms ANCHOR, LDA, and SLDA on all three datasets. We report the results based on LOGISTIC as it produces the best accuracy consistently for ANCHOR, SUPANCHOR, and LDA.

from ANCHOR. Similarly, with twenty topics on the YELP dataset, SUPANCHOR has 0.77 accuracy while ANCHOR has 0.74. Our SUPANCHOR model is able to incorporate metadata to learn better representations for predicting sentiment. Moreover, in Section 4.5.2 we show that SUPANCHOR does not need to sacrifice topic quality to gain predictive power.

More surprising is that SUPANCHOR also outperforms SLDA. Like SUPANCHOR, SLDA jointly learns topics and their relation to metadata such as sentiment. Figure 4.5 shows that this trend is consistent on all sentiment datasets. On average, SUPANCHOR is 2.2 percent better than SLDA on AMAZON, and 2.0 percent better on both YELP and TRIPADVISOR. Furthermore, SUPANCHOR is much faster than SLDA.

SLDA has lower accuracy than SUPANCHOR in part because SUPANCHOR jointly finds specific lexical terms that improve prediction. Forming anchor words around the same strong lexical cues could discover better topics. In contrast, SLDA must discover the relationship through the proxy of topics. Similar results are observed in [Nguyen et al., 2013c, SHLDA] that jointly model topic-based and lexical-based parameters or in Ramage et al. [2010] that interpolate topic-based features and lexical-case features.

4.4.3 Combination of Topic Features and Lexical Features

Even though we care about topic representation learned by topic models for sentiment classification, lexical features such as word n-gram or TF-IDF have always dominated text classification [Furnkranz, 1998]. Ramage et al. [2010] show that interpolating topic and lexical features often provides better classification than either alone. Here, we take the same approach and show how different interpolations of topic and lexical features create better classifiers. We first select an interpolation value λ in $\{0, 0.1, 0.2, \dots, 1\}$, and we then form a new feature vector by concatenating λ-weighted topic features with $(1-\lambda)$-weighted lexical features. Figure 4.6 shows the interplay between topic features and TF-IDF features as the weight of topic features increases from zero (all TF-IDF) to one hundred (all SUPANCHOR topic features) percent on the AMAZON dataset (other

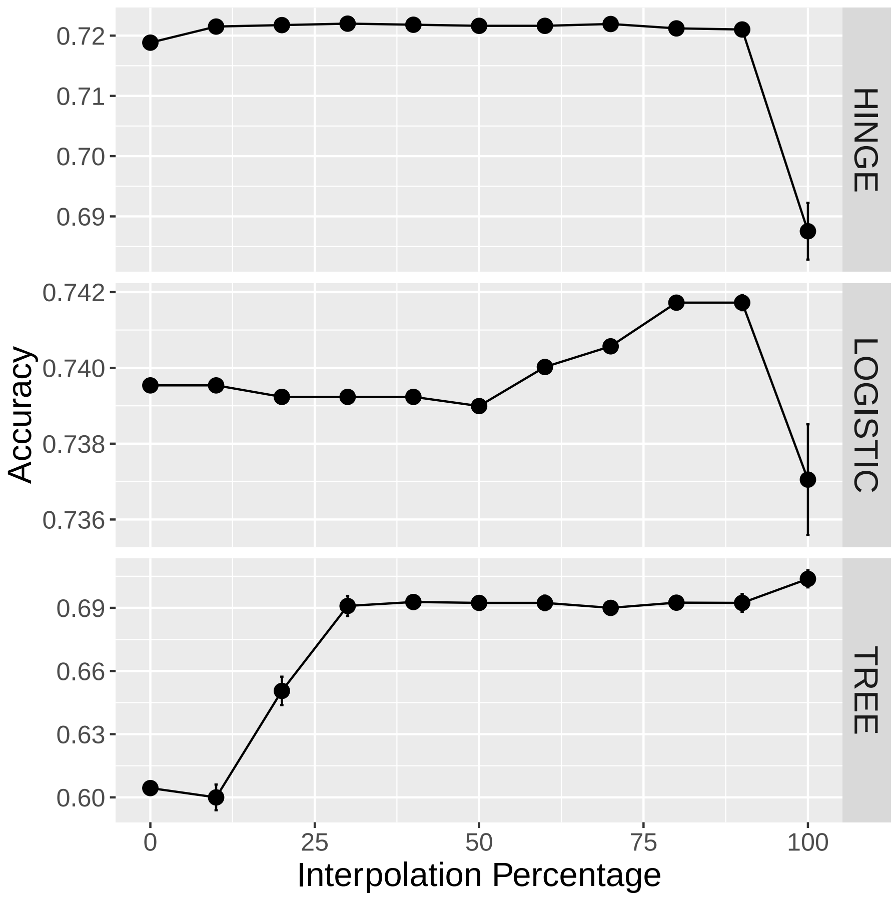


Figure 4.6: Mean accuracy on AMAZON with eighty topics using five random runs. Error bars indicate 95% confidence intervals. SUPANCHOR produces good representations for sentiment classification that can be improved by interpolating with lexical TF-IDF features. The interpolation (x-axis) ranges from zero (all TF-IDF features) to one hundred (all SUPANCHOR topic features). For example, in the case of HINGE, combining 10 percent of topic features with 90 percent of lexical features improves prediction accuracy over using lexical features alone (the 95% confidence interval of this difference is $[0.00226, 0.00307]; t = 15.093, p < 0.0001$).

datasets are similar).[7] Combining both feature sets is better than either alone in most cases, although the interpolation depends on the classifier. We observe that for the best classifiers, LOGISTIC, combined features clearly show their advantages. For the HINGE classifiers, combined features consistently outperform either lexical or topic features alone, however, the gain is pretty small (about 0.25%). For the TREE classifiers, which perform poorly,

[7] As before, we do parameter selection using cross-validation on TRAIN data and report final TEST results. We report the mean accuracy using five random runs of the SUPANCHOR. For combined features, we observe very small variation in the accuracy scores across all classifiers.

adding lexical features degrades accuracy performance. These results show the value of good topic features for classification task; they complement lexical features to achieve best possible performance.

4.5 Sentiment Anchor Words: New Insights for Understanding Text

Since we augment the word representation with word and sentiment co-occurrence, the supervised anchor word algorithm learns additional anchor words that reflect sentiment topics. These sentiment anchor words play two important roles: (1) explaining the SUPANCHOR's predictive power and (2) providing insights to understand sentiment documents.

4.5.1 Sentiment Topic Words

As mentioned above, the topics produced by the ANCHOR and SUPANCHOR algorithms have many similarities. In Figure 4.7, nearly all of the anchor words discovered by ANCHOR are also used by SUPANCHOR. These anchor words tend to describe general food types, such as "pizza" or "burger", and characterize the YELP dataset well. The similarity of these shared topics explains why both ANCHOR and SUPANCHOR achieve similar topic interpretability scores (see Section 4.5.2).

To explain the predictive power of SUPANCHOR we must examine the anchor words and topics unique to both algorithms. The anchor words which are unique to ANCHOR include a general topic about wine and two somewhat coherent topics related to time. By adding supervision to the model, we get three new anchor words (to replace old ones) which identify sentiment ranging from extremely positive reviews mentioning a favorite

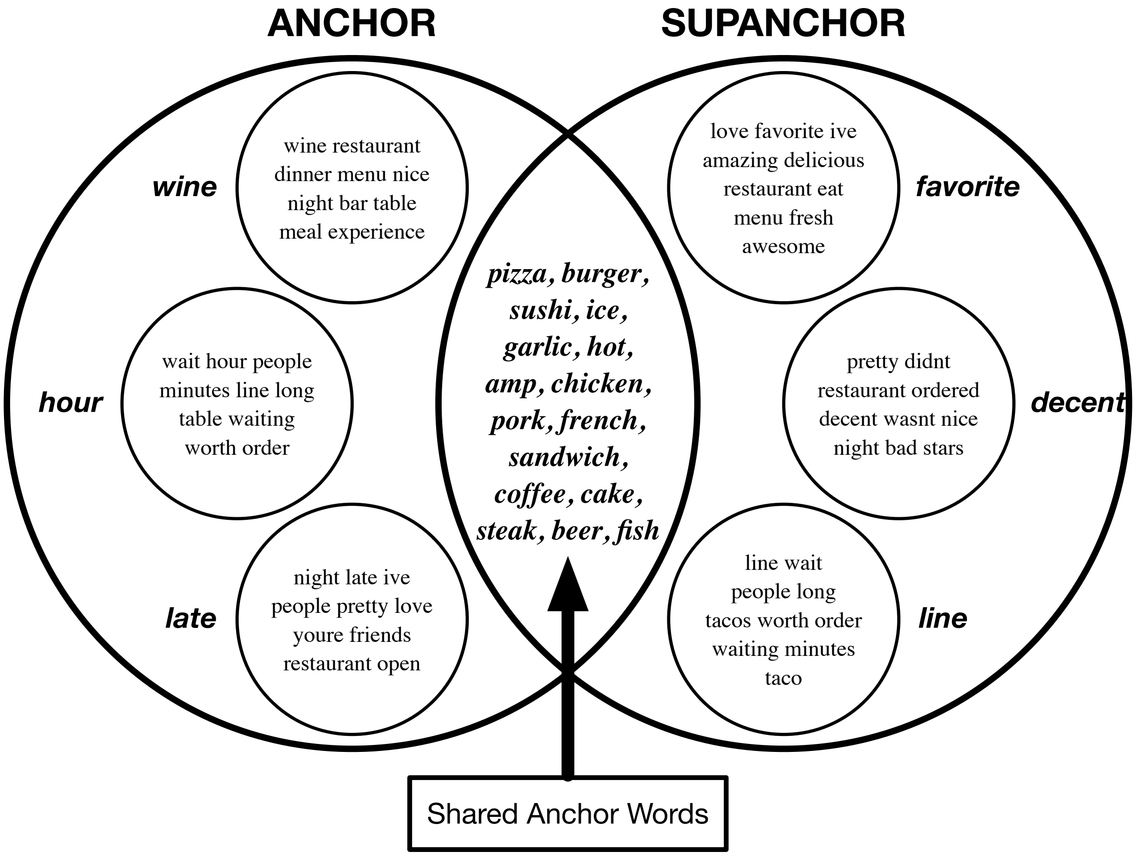

ANCHOR
SUPANCHOR
wine
wine restaurant dinner menu nice night bar table meal experience
hour
wait hour people minutes line long table waiting worth order
late
night late ive people pretty love youre friends restaurant open
pizza, burger, sushi, ice, garlic, hot, amp, chicken, pork, french, sandwich, coffee, cake, steak, beer, fish
favorite
love favorite ive amazing delicious restaurant eat menu fresh awesome
decent
pretty didnt restaurant ordered decent wasnt nice night bad stars
line
line wait people long tacos worth order waiting minutes taco
Shared Anchor Words

Figure 4.7: Comparing topics generated for the YELP dataset: anchor words shared by both ANCHOR and SUPANCHOR are listed. Unique anchor words for each algorithm are listed along with the top ten words for that topic. For clarity, we pruned words which appear in more than 3000 documents as these words appear in every topic. The distinct anchor words reflect positive ("favorite") and negative ("line") sentiment rather than less sentiment-specific qualities of restaurants (e.g., restaurants open "late").

restaurant to extremely negative reviews complaining about long waits. Capturing more sentiment specific anchor words to replace general anchor words shows strong effect of augmenting sentiment dimensions on how anchor word algorithms choose anchor words and hence generate new sentiment topics.

This general trend is seen across each of the datasets, providing us more understanding about topics and documents. For example, ANCHOR and SUPANCHOR both discover shared topics describing consumer goods, but SUPANCHOR replaces two topics discussing headphones with topics describing “frustrating” products and “great” products (in the AMAZON dataset). Similarly, in the TRIPADVISOR data, both ANCHOR and SUPANCHOR share topics about specific destinations, but only SUPANCHOR discovers a topic describing “disgusting” hotel rooms.

4.5.2 Topic Interpretability

Similar to Section 3.5.3, we evaluate the quality of topics produced by each model using topic interpretability (Section 2.3.2). We used half a million documents from Wikipedia as a proxy corpus to compute the induced normalized pairwise mutual information (NPMI) on the top ten words in topics as a proxy for interpretability.

Figure 4.8 shows the NPMI scores for each model. Unsurprisingly, unsupervised models (LDA) produce the best topic quality. In contrast, supervised models must balance metadata (i.e., response variable) prediction against capturing topic structure. Consequently, SLDA does slightly worse for topic interpretability.

SUPANCHOR and ANCHOR produce similar topic quality on all datasets. Since SUPANCHOR and ANCHOR have nearly identical runtime, SUPANCHOR is better suited for supervised tasks because it improves classification without sacrificing interpretability. It is possible that regularization would improve the interpretability of these topics; as shown in Chapter 3, adding regularization removes overly frequent words from anchor-discovered topics.

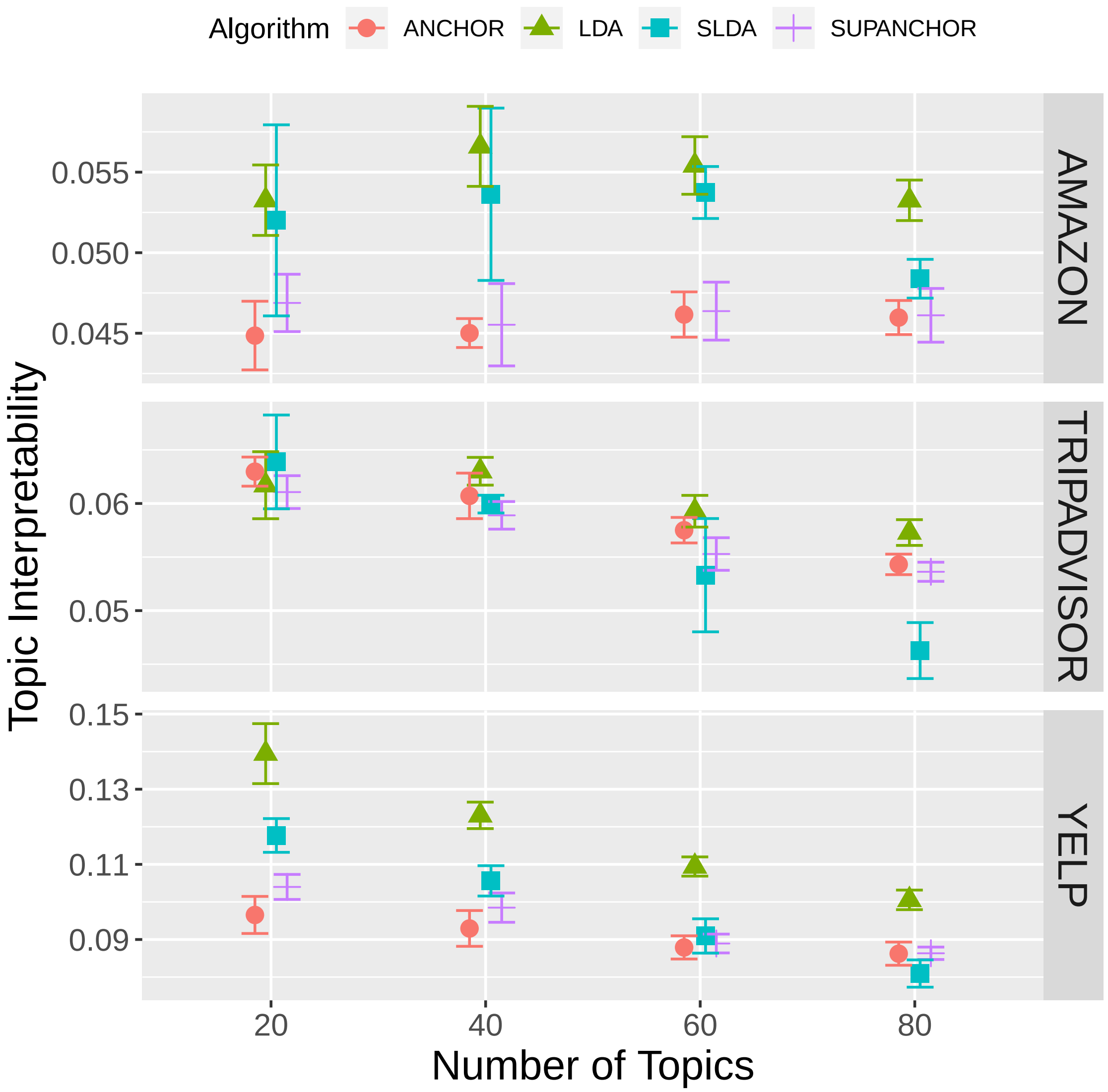


Figure 4.8: Mean topic interpretability for all three datasets. Error bars indicate 95% confidence intervals. SUPANCHOR and ANCHOR produce the same topic quality. LDA outperforms all other models and produces the best topics. Performance of SLDA degrades significantly as the number of topic increases.

4.6 Runtime Analysis

Having demonstrated that SUPANCHOR outperforms both ANCHOR and SLDA, this section shows that SUPANCHOR also inherits the runtime efficiency from ANCHOR. Table 4.2 summarizes the runtime of all models for both AMAZON and TRIPADVISOR on a six-core

Dataset	Measure	SUPANCHOR	LDA	SLDA
AMAZON	Preprocessing	32	32	32
	Generating $\bar{Q}/S$	29		
	Training	33	886	4,762
	LDAC inference	38 (train), 13 (dev/test)		
	Classification	<5	<5	
TRIPADVISOR	Preprocessing	305	305	305
	Generating $\bar{Q}/S$	262		
	Training	181	8,158	71,967
	LDAC inference	830 (train), 280 (dev/test)		
	Classification	<5	<5	

Table 4.2: Runtime statistics (in seconds) for the AMAZON and TRIPADVISOR datasets. Blank cells indicate a timing which does not apply to a particular model. SUPANCHOR is significantly faster than conventional methods.

2.8GHz Intel Xeon X5660. On the small dataset AMAZON, SUPANCHOR completes the training within one minute, and for the larger TRIPADVISOR dataset, it completes the learning in approximately three minutes. The main bottleneck for SUPANCHOR is learning the document distributions over topics (topic features for classification), although even this stage is fast for known topic distributions. This result is far better than the twenty hours required by SLDA to train on TRIPADVISOR.

4.7 Upstream or Downstream?

Supervised topic models that are used for incorporating side information, such as labels, are often classified into *upstream* or *downstream* classes of models. In the case of upstream models, the label information is conditioned to generate the latent topics; in short, these models assume that labels are generated first, and then topics. Examples include the DiscLDA [Lacoste-Julien et al., 2009] and the Dirichlet-multinomial regression (DMR) [Mimno and McCallum, 2008]. In the downstream models, latent topics are generated first and are then used to predict the label information. Examples of downstream models are MedLDA [Zhu et al., 2009a] and SLDA [Blei and McAuliffe, 2007].

Recently, Arora et al. [2019] categorize the techniques that use unlabeled data to map each original data point into a feature vector as *contrastive learning* and show theoretical results on why such techniques are working for downstream classification tasks. For classification, unsupervised topic models such as LDA and ANCHOR are downstream models; they belong to contrastive learning because they learn the topic features of documents which are used by classifiers to predict labels.

In SUPANCHOR, we enrich word vector representation with supervision to learn new anchor words (Figure 4.2), which in turn helps us to recover new sets of topics. We can intuitively interpret that SUPANCHOR is an upstream model because topics are generated depending on words and labels. However, SUPANCHOR can also be interpreted as a downstream model, because SUPANCHOR is also like ANCHOR, in which we use the topic features of documents with a classifier to make predictions for labels. Technically, the concept of *upstream* or *downstream* models often applies only to probabilistic topic models where there is a clear generative process for how data are generated. For spectral models, such as the anchor word algorithms, where models recover the latent topics to explain observed data, it is difficult to distinguish clearly between upstream and downstream because there is no generative process involved. Arriving at a conclusive distinction requires theoretical formalism, which is worthy of future work.

4.8 Conclusions

The supervised anchor word algorithm provides a general framework for learning highly interpretable topic representations by taking advantage of both word co-occurrence and metadata. Our straightforward extension (Equation 4.1) places each word in a vector space that not only captures co-occurrence with other terms, but also the interaction of the word and its sentiment, in contrast with algorithms that only consider raw words.

Moreover, the supervised anchor word algorithm is fast: it inherits the polynomial time efficiency from the unsupervised anchor word algorithm. It is also effective: it is better at providing features for classification than unsupervised topic models and also better than supervised topic models with conventional probabilistic inference such as SLDA.

Our supervised anchor word algorithm offers the ability to analyze datasets without the overhead of Gibbs sampling or variational inference, allowing users to quickly interpret big data and make decisions accordingly. Combining bag-of-words analysis with metadata through efficient, low-latency topic analysis allows users to quickly obtain deep insights.

One limitation of the supervised anchor word algorithm results from an inherent property of the original anchor word algorithm, which is that it does not directly produce the topic features of documents (the document topic matrix θ), and we must still use sampling techniques to estimate the topic features of documents after recovering topic distributions.

Another limitation of the supervised anchor word algorithm is its inability to directly make predictions (as in the case of SLDA). Instead, it requires a classifier (e.g., SVM) to make predictions using topic features. Using externally supervised learning models may slow down the training step if these models require significant hyper-parameter tuning.

One assumption in the SUPANCHOR model is the discovery of new label-specific anchor words. Without finding label-specific anchor words, the supervised anchor word algorithm operates exactly like the regular anchor word algorithm, because the topic coefficient recovery step (Equation 2.29) is unmodified. Although we do not observe this case in our empirical experiments, understanding the true behavior of how label-specific anchor words are learned is critical for designing future supervised models based on the anchor method.

Identifying flexible topic models to address various types of datasets has been one of the main goals in this thesis. A benefit of the supervised anchor word algorithm that contributes to the variability dimension is its flexibility to incorporate various types of supervised information. For instance, anchor word representations combined with word embeddings [Mikolov et al., 2013a, Pennington et al., 2014] that are learned from large unlabeled and labeled datasets can capture external sources of knowledge (see Section 1.2). While our experiments focus on binary classification, the same technique is also applicable to multi-class and multi-label classification. In the next chapter, we explore the supervised anchor word algorithm for the multi-class classification problem within the context of active learning.

Chapter 5

The Suitability of Supervised Topic Models for Active Learning

5.1 Introduction

In the previous chapter, we introduced a high-speed and accurate supervised topic model based on the anchor word algorithm. However, our experiments were conducted on corpora in which we assumed that all training documents had their associated labels, and therefore that the supervised topic models could fully learn topics and their associations with labels to best support prediction. In many practical, real-world problems, however, we do not have this luxury, and we often have to adjust our models to learn from a much smaller set of labeled documents (rather than from all available documents). Fortunately, most supervised topic models such as SUPANCHOR or SLDA can perform satisfactorily even under these constraints. In this chapter, we investigate the idea of incorporating supervised topic models into active learning—a subfield of semi-supervised learning in which learning algorithms are designed to work with a small set of labeled training data points but still take advantage of the huge reserve of unlabeled data points. The key idea in active learning strategies is to use an active learner to iteratively collect labels for the most useful data points.

Active learning reduces the amount of human annotation effort needed to generate metadata (e.g., labels), which are required to train a supervised learning algorithm [Tong and Koller, 2002, Settles, 2010, 2012]. Active learning uses a querying strategy to identify and request metadata for those data points that are most beneficial for training a supervised learning model. One immediate benefit of this approach, for example, is that it reduces the number of training data points needed. Several querying strategies, such as uncertainty sampling [Lewis and Gale, 1994a] and query by committee [Seung et al., 1992b], have shown practical success [McCallum and Nigam, 1998, Settles and Craven, 2008b, Settles, 2011b]. When we use active learning for natural language, the data point of interest is a document in the corpus. For the rest of this thesis, we will use *document* in the place of *data point* when we explain active querying strategies.

Using active learning strategies, the predictive model trained on the current training set selects the next document to query for labeling. The criteria for selecting that next document depend on which active learning strategies are being used. Document selection criteria are based on information at both the document level and the overall corpus level, together with the current predictive accuracy of the classifier at the moment of training. Therefore, we hypothesize that providing the model with more *insights* (e.g., information to discriminate documents such as how each document talks about some topics) on the dataset will lead to more informative queries and thus will create a better training set more quickly. When the goal is to classify documents, topic models such as LDA provide an overview of the corpus. The classifier can use this information to pose more effective queries. This idea is not entirely new and is related to exploiting the cluster structure in data as thoroughly explained in Dasgupta and Hsu [2008]. Furthermore, a topical overview of the corpus can help annotators (or users) infer a global label set in a scenario for which labels are unknown (e.g., tracks in a conference) [Poursabzi-Sangdeh et al., 2016]. Because supervised topic models learn topical insights that are predictive for *labeled* documents, we use supervised topic models to update topics incrementally as new labeled documents become available.

Using supervised topic models for active learning comes with two challenges. The first comes from the interactive nature of active learning—that is, the involvement of humans in the document labeling process. This is the *latency* issue, where human annotators ask for quick update from the model, in contrast to the *throughput*, which depends on how fast a human annotator works. To ensure that topics are quickly updated and provided to the classifier, supervised topic models need to run very fast. In this respect, the supervised anchor word topic model from the previous chapter is an ideal candidate. We further improve its inference step to make it run even more quickly for enhanced interactivity. The second challenge comes from how we incorporate updated topics so that information is refreshed and effectively used to improve the model. We address this issue by proposing new querying strategies based on global document overviews using topic models.

In the next sections, we formally introduce active learning and several popular active learning query strategies. We carefully review the uncertainty sampling strategy, which we then adapt by combining document-level prediction probability vectors produced by the classifier with document-level topic vectors produced by topic models to create various new querying strategies. Section 5.2 presents a pipeline for incorporating supervised topic models into active learning. To enhance user interactivity, we devise a fast inference approach for SUPANCHOR to quickly estimate document topic distributions using matrix multiplication. We then conduct experiments using three multi-class labeled datasets—20 Newsgroups, U.S. Congressional Bills, and Reuters—to confirm the feasibility of providing additional features to the active learner and classifier via supervised and unsupervised topic models. In Section 5.3, we show that our proposed active learning strategies incorporating document overviews and updated topics improve model performance in comparison to the framework deployed by Poursabzi-Sangdeh et al. [2016], which uses only static topics from LDA. We draw chapter conclusions in Section 5.4.[1]

5.1.1 What is Active Learning?

Active learning is a subfield of machine learning; it is considered a type of semi-supervised learning. The active learning framework is built on the concept that if a learning algorithm can choose which data points to learn from, it can significantly improve its *learning capability*.[2] To improve the learning capability of any learning algorithm, researchers work to enhance one of two main properties. The first property focuses on the ability of a learning algorithm to produce better performance, for example, prediction accuracy. The second property focuses on the number of training data points required for the learning algorithm to achieve a certain level of performance.

So why is active learning useful? In short, because it can enhance the learning capacity of a learning algorithm. In practice, we tend to use supervised learning algorithms only if they can provide a reasonable level of performance (e.g., prediction accuracy greater than 80%). There is no concrete rule, however. Instead, it is based on the real-world use case, and the acceptable value is often defined by users. For some users, 80% is great, but for others, it is effectively useless. To achieve an acceptable level of performance, we often have to use a very large training dataset with thousands or even millions of data

[1] The author did all the work in this chapter.

[2] In theoretical machine learning, the capacity of a model is captured by the concept of VC dimension [Vapnik et al., 1994]; however, in this thesis we focus on the practical usage of a model and measure its capacity through its performance on tasks.

points. Collecting labels for data points, however, is extremely difficult and very costly. A framework such as active learning can reduce the number of data points the learning algorithm needs.

A typical active learning framework is iterative and contains the following steps.

1. Randomly select a small number of data points, and ask users to label.

2. Train a fast classifier (e.g., logistic regression) to learn those labeled data points. For each unlabeled data point, build a probability vector that reflects how likely it is that the data point belongs to a given class.

3. Use a query strategy to choose the next data point from the pool of all data points.

4. Present this selected data point to the user, and ask for a label. Add the newly labeled data point to the set of previously labeled data points.

5. Repeat above steps. Figure 5.1 illustrates this iterative process.[3]

Active learning works because it focuses on capturing quality labeled data points—those that the learning algorithm can extract the most information from. For concreteness, let us focus on a simple case of binary classification where our goal is to filter out spam emails. We follow Settles [2010] and demonstrate the value of quality data points in Figure 5.2. In this demonstration, a linear classifier finds the decision boundary that separates two classes of labels: squares and circles. Data points that are closer to this boundary are more important than the other data points; this small subset of data points is

[3] Note also that, instead of posing just one data point at a time, we can provide a small set of data points and ask users to label.

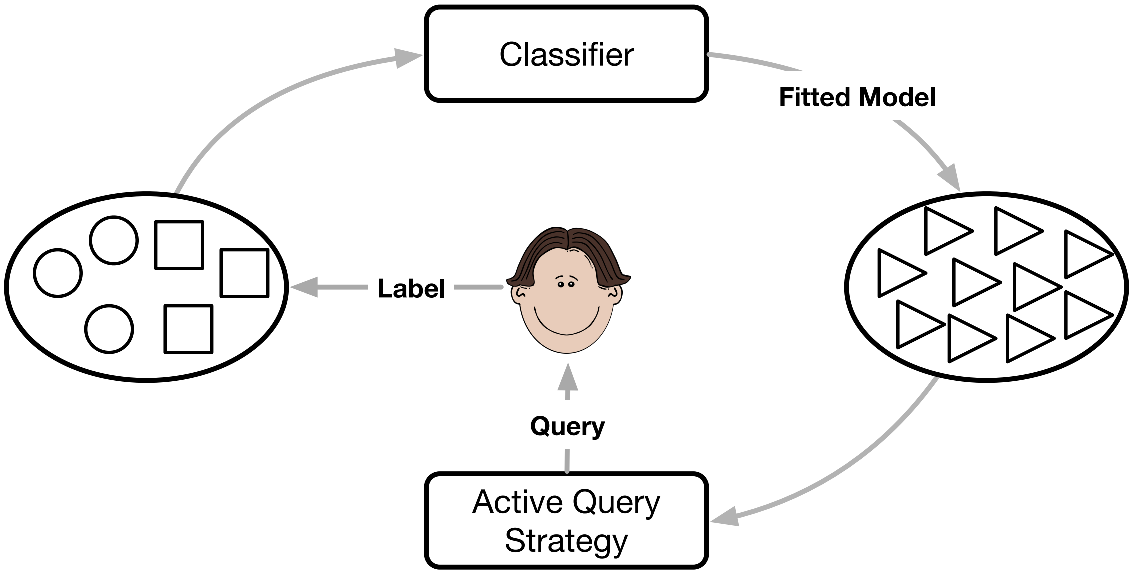


Figure 5.1: Iterative process in active learning in which an active learner continually queries users for labels. We demonstrate this process using a binary classification task of filtering spam emails: spam emails are squares and legitimate emails are circles. Starting with a dataset of unlabeled data points (triangles, right oval), an active learner iteratively applies its querying strategy to select data points for a user to label (left oval). The updated set of labeled data points becomes a training set for a classifier to train on. The classifier then applies what it has learned to unlabeled data points.

sufficient for the classifier to use in finding the decision boundary. Therefore, identifying those quality data points is critical.

Active learning relies on query strategies to probe for these quality data points. Different query strategies have different definitions of quality. Let us first review some of the strategies where the data points are documents.

5.1.2 How to Query a Quality Document?

Informativeness of Documents In supervised learning, the classifier C, while predicting a label for a document d, assigns a probability distribution vector over all labels ys, and we can use this distribution vector to measure the *entropy* over labels of that document [Shannon, 2001]: $\mathbb{H}_C\left[y_d\right] = -\sum_i^L P(y_i \mid d)\log(P(y_i \mid d))$, where L is the number of

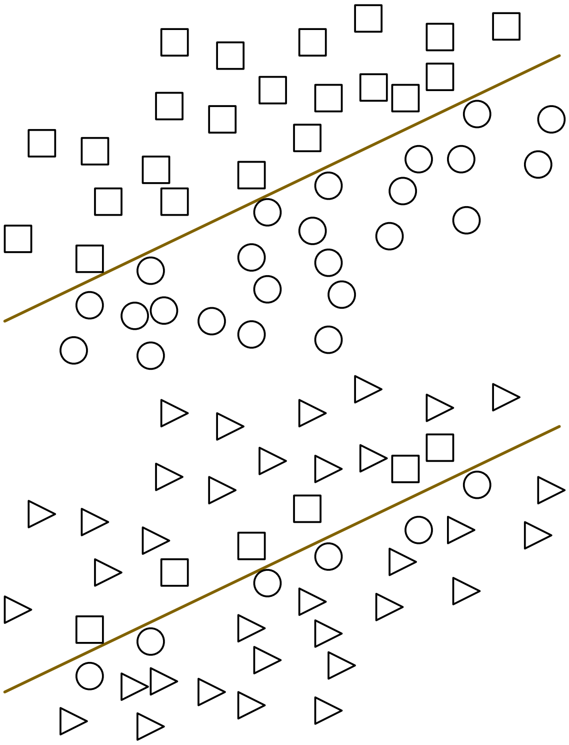


Figure 5.2: Demonstration to show how active learning helps a spam filtering classifier (a supervised learning task).We use *squares* for spam emails and *circles* for legitimate emails. By posing quality data points, we can train the classifier using fewer labeled data points (bottom) while still producing a level of accuracy similar to what we would get if we had labeled all data points (top).

possible labels. Entropy is an information-theoretic measure that represents the amount of information needed to encode a distribution. A high entropy value means that the classifier is confused about which label a document belongs under; in contrast, a low entropy value means that the classifier is more confident about which label this document belongs under. So, a classifier learns more information when a high-entropy document is labeled than when a low-entropy document is labeled. Hence, a higher-entropy document has a more informative label for the classifier to learn from. Going back to the concept of *quality* documents that we mentioned in the previous section, obviously, we can equate high-entropy documents with quality documents, because the label of an informative document gives the classifier more information about the document collection than it would get from a lower-entropy (i.e., less informative) document. By consolidating the concept of quality with informativeness (or entropy), we have found an excellent way to quantitatively define quality documents in active learning. This viewpoint, indeed, motivates much work in active learning research on querying strategies. Let us describe several popular active learning querying strategies.

Uncertainty Sampling Uncertainty sampling [Lewis and Gale, 1994b] is a special case of pool-based sampling [Lewis and Gale, 1994b, McCallum and Nigam, 1998, Hoi et al., 2006, Tong and Koller, 2002] in which an active learner presents the unlabeled document that it is least certain about (that is, the one that will be most informative once we learn its label). This approach is very straightforward; at every iteration of the cycle, the active learner will present the document with the highest entropy value.

Query-by-Committee The idea of query-by-committee (or QBC) is that an active learner maintains a list of learning models called *the committee* [Seung et al., 1992a]. After these learning models have completed training on the labeled dataset, each model is able to assign labels to unlabeled documents. QBC works by selecting the unlabeled document that generates the highest level of disagreement among these committee models and presents that document to the user for labeling. There are two items in the QBC approach: a set of learning models with different hypotheses and a method to measure the level of disagreement among the models.

The above querying strategies focus only on the use of already-labeled documents

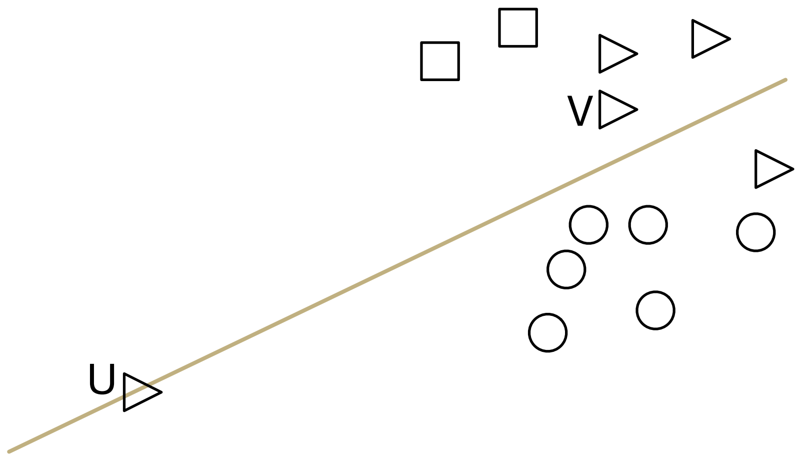


Figure 5.3: Illustration of the failure of uncertainty sampling when it tends to pick an outlier document (U) instead of (V). Even though U is on the boundary, it is still very far away from other documents, which makes it an outlier. This situation should be avoided, because U does not contain useful information for classification.

to create a way to query for a label for an unlabeled document based on how uncertain a classifier is. It is natural to ask how we can utilize the unlabeled dataset to create effective querying strategies. The concept of *representativeness* does just that. Besides the informativeness of the selected document, active learning strategies use the concept of representativeness to select a document that best represents the overall unlabeled dataset. We investigate a technique by Settles and Craven [2008a] that relies on similarity among documents to choose a representative document.

General Density-Weighting Technique Most querying strategies such as uncertainty sampling or query-by-committee focus primarily on local information in individual documents rather than on the entire document space, making the model more prone to querying outliers. Figure 5.3 demonstrates this case for a binary classification problem using the uncertainty sampling technique. The document chosen has the highest entropy value; however, it is not a good representative for other documents in the corpus. An example of such document is an email that contains many rare spam trigger words, such as “blockchain dollars”—spam words that were not observed frequently before. To remedy this issue, we can explicitly model the input space of documents during the querying strategy [Settles, 2012].

Settles [2008], Settles and Craven [2008a] introduce a general technique based on information density called the *general density-weighting technique*. This technique combines two important concepts: informativeness and representativeness. According to this technique, a selected document should be not only informative but also representative of the rest of document collection—this reduces the chance of it being an outlier. As described above, informativeness is a measure of the document’s entropy as given by the classifier of choice, whereas the representativeness of a document is measured in many different ways. Researchers have proposed various techniques to measure representativeness that use either clustering or raw document representation as a vector to compute distance [Settles, 2012, Nguyen and Smeulders, 2004, Xu et al., 2007, Settles and Craven, 2008a].

The general formula given by Settles and Craven [2008a] for how to query the next document is

$$d^* = \arg\max_d \left(\mathbb{H}_C\left[y_d\right] \times \left(\frac{1}{D} \sum_{i=1}^{D} \text{similarity}(d, d^{(i)}) \right)^p \right), \tag{5.1}$$

where the first term is the normal entropy and the second term measures the average similarity of the document d to all other documents, with p controlling its relative importance. The second term is expensive to compute fully, therefore clustering or pre-computation techniques are used for interactive scenarios [Settles and Craven, 2008b].

The above techniques can be summarized as trade-offs between informativeness and representativeness. Most active learning algorithms use only one of these criteria. Earlier models, for example, tended to use informativeness-focused techniques such as uncertainty sampling or query-by-committee, whereas newer techniques focus more on representativeness by using more information from unlabeled documents [Li et al., 2012]. Either of the criteria can produce good models, but they often have limitations such as relying on outliers or depending heavily on the clustering model. In this thesis, we combine informativeness from traditional measures of document entropy with representativeness learned by running topic models.

The introduction of active learning has motivated researchers to invent new approaches to ensure that the active learner of choice can best understand unlabeled documents before suggesting the next document for a label. For a complete review of other querying strategies, such as expected error reduction [Roy and McCallum, 2001, Guo and Greiner, 2007, Moskovitch et al., 2007] (and variance reduction methods) or expected model change [Settles et al., 2008], please refer to Settles [2010].

5.2 Supervised Topic Models for Active Learning

Poursabzi-Sangdeh et al. [2016] propose active learning with topic overviews (ALTO), which is an interactive framework that combines uncertainty sampling with LDA to reduce the annotation effort in classifying documents and which is particularly helpful in preparing the label set. The main motivation for the ALTO framework comes from the challenge of annotating documents: this task requires users to have both *global* and *local* knowledge of the entire dataset. Local knowledge allows annotators to label the documents correctly, and global knowledge helps annotators to create the set of overall labels (an overview of the entire corpus). ALTO has two main steps: (1) run LDA on the whole collection of documents before annotation and (2) present top documents within each topic to users for annotation. The latter step is similar to the typical, iterative active learning process presented previously. The key difference is that instead of presenting best documents one at a time, ALTO presents a set of topics and documents grouped under each topic to give annotators more of an overview to better capture new labels.

ALTO is also slightly different from traditional active learning techniques in that it grows the label set as more documents get labeled. Despite this difference, ALTO should work in a traditional active learning setting where the set of labels is fixed, and as more documents get their labels, the classification accuracy should be improved. We focus on the active query strategies that ALTO uses. Specifically, given a corpus overview coming from LDA, ALTO improves upon uncertainty sampling by combining the document-label entropies produced by the classifier with the document-topic probabilities produced by LDA to present the best document from the unlabeled set of remaining documents. Using ALTO, we can deploy the two following active learning strategies based on uncertainty sampling.

Most Popular Topic One of the main contributions of the ALTO framework is the combination of document-label entropies with document-topic probabilities, in which the probability of the most dominant topic for each document contributes to the probability that an active learner will select that document

$$d^* = \arg\max_{d} \left(\mathbb{H}_C \left[y_d \right] \theta_{d,k_d} \right), \tag{5.2}$$

where $\mathbb{H}_C \left[y_d \right] = -\sum_i P(y_i \,|\, d) \log P(y_i \,|\, d)$ is the document-label entropy, and θ_{d,k_d} is the probability of the most prominent topic $k_d = \arg\max_k \left(\theta_{d,k} \right)$ in document d. This strategy poses queries on documents when the classifier is uncertain about their label *and* they are representative of a topic. This intuition is actually inspired by the work of Dasgupta and Hsu [2008] using hierarchical clusters to balance document coverage and the classifier accuracy; the difference is that topics produced by LDA are *flat* clusters. In its interactive annotation framework, ALTO shows the annotators K (the number of topics) groups of documents to provide a corpus overview. By running a user study using this framework, Poursabzi-Sangdeh et al. [2016] showed that ALTO guides the annotators to focus more on diverse sets of documents (because presented documents are representative of a topic) and also helps with the quality of the label set in the user study. However, ALTO was not evaluated using traditional active learning methods such as assessing how accurate it is for a predictive task. The ability to induce a correct label set that matches the gold label sets [Poursabzi-Sangdeh et al., 2016] is useful, but it does not guarantee that the set of annotated documents is an optimal training set for the classifier to classify documents—the ultimate goal of active learning when the number of unlabeled documents is huge. This chapter evaluates the predictive effectiveness of ALTO and also improve it by combining it with supervised topic models.

Topic Entropies Instead of using the dominant topic within each document, we introduce in this chapter another strategy to select the most informative document when topics within a document are difficult to distinguish:

$$d^* = \arg\max_d \left(\mathbb{H}_C \left[y_d \right] \times \mathbb{H}_C \left[\theta_d \right] \right). \tag{5.3}$$

This strategy poses queries on documents when the classifier is uncertain about both their label and their topic. The intuition behind this strategy is capture the most confusing document for both topic model and classifier.

What is ALTO *missing?*

The ALTO framework provides a global overview of the data using topics generated by LDA. Because LDA is run before the experiment, however, the topic model does not change as documents are labeled incrementally. The use of static topics produced by LDA is useful for the initial rounds of the annotation process, but it becomes increasingly limited as more and more labeled documents are collected. The global overview through topics is only useful for users if this view captures not only the general themes of the corpus but also themes that are associated with labels. The supervised topic models discussed in Chapter 4 produce an updated view of the corpus from *label*-oriented topics, and this is exactly the view we need for active learning. In the next section, we discuss how to incorporate supervised topic models into an ALTO's active learning framework.

5.2.1 Query Strategies with Updating Topics

We investigate the effect of updating the topic model on the prediction performance of active learning strategies in scenarios in which the label set is known. Our approach differs from ALTO in two ways. First, we assume the label set is known, thus the label provided by the annotator is always correct. In contrast, ALTO assumes that the label set is unknown, and annotators need to induce the set of labels as they perform the annotation task. The second distinction is our use of supervised topic models to update topics. Instead of using static topics from LDA as in ALTO, we use dynamic topics from supervised topic models as new labeled documents come into the system. These dynamic topics provide an updated global overview of both the general corpus and the labeling structure, guiding users more effectively with higher-quality information.

By using dynamic topics, we can create many query strategies for active learning. For example, simply expanding ALTO would mean incorporating supervised topic models into active learning by using the topic distributions of documents as features for a classifier and measuring the weighted entropy as in Equation 5.2. Unlike ALTO, however, where θ stays static, our supervised topic model updates θ dynamically to capture newly available metadata information (see Section 2.4.2 and Section 4.5.1).

We start with an initial set of randomly selected labeled documents, then iteratively train a supervised topic model to obtain the updated θ matrix, and finally use Equation 5.2 to pose a query for the label of the next document. Note that all of these strategies focus only on the individual document level (the informativeness of a document) without considering the relationship between the selected document and other documents (the

Algorithm 4 Supervised Topic Models for Active Learning

Input: A collection of unlabeled documents UC and a small, initial set of labeled documents LC. A supervised learning model SLM.
Output: A sufficient LC after T iterations.

```
Run SUPANCHOR using labeled documents from LC and unlabeled documents from UC.
  Produce $\theta$ matrix where $\theta_d$ is the document-topic vector for the document $d$.
while iter $<$ T do
    Train SLM using document labels from LC and $\theta$ matrix for document features. Produce an
      entropy matrix $\mathbb{H}_C$ where $\mathbb{H}_C\,[y_d]$ is the classifier entropy for the document $d$.
    Use the Equation 5.2 to select the best document $d^*$.
    Ask user to label $d^*$.
    Update LC with $d^*$ and its label.
    Rerun SUPANCHOR.
return Final LC of labeled documents.
```

representativeness of a document).

One of the challenges of incorporating supervised topic models into active learning is the interactive nature of the active learning framework, which requires low latency in order to interact with humans. In the previous chapter, we saw the superior performance in both speed and accuracy, of the SUPANCHOR algorithm in comparison to other topic models such as SLDA; therefore, we focus on the SUPANCHOR algorithm for active learning. Our new algorithm is shown in Algorithm 4, using Equation 5.2 as the querying strategy. In the next section, we describe a fast version of our SUPANCHOR algorithm that is more suitable for active learning due to its higher speed.

5.2.2 Speeding Up for Interactivity: Instant Inference Supervised Anchor

LDA-based supervised topic models such as SLDA update topics through expensive probabilistic inference schemes (see Section 2.2) that are slow. Given that our experiments are motivated by an interactive framework where human annotators are asked to label

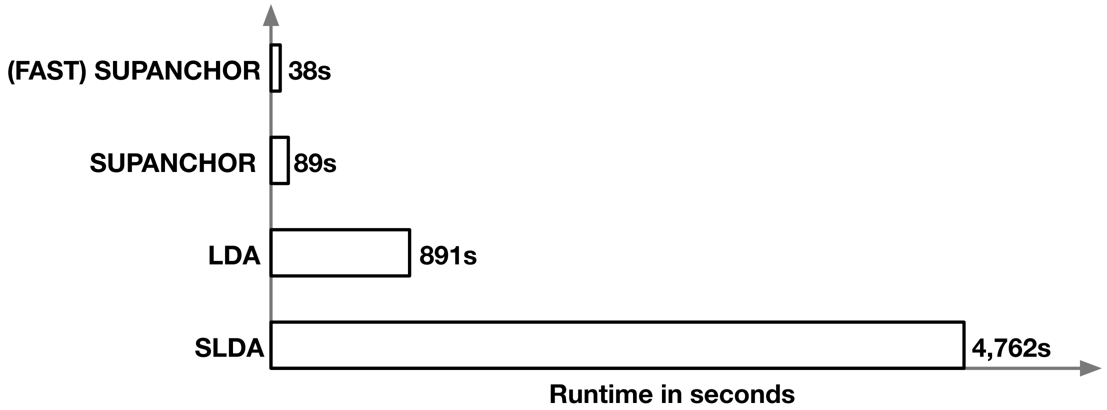


Figure 5.4: Benchmarking using 80 topics and 13,300 documents from the AMAZON dataset (Chapter 4): a fast inference version of SUPANCHOR runs significantly faster than other topic models.

documents, high latency makes these models impractical. Therefore, we adopt the fast supervised anchor word algorithm in our experiments.

Inferring topics using SUPANCHOR is fast, because unlike SLDA, which operates on *tokens*, SUPANCHOR operates on *word types*. However, a direct inference of the θ distribution is infeasible. As we have seen in Section 3.5, we use Gibbs sampling to recover the document-topic distributions θ, and this sampling step is usually the bottleneck. Recall that the J matrix introduced in Section 2.5.1 is the observed term-document matrix of dimensions $V \times D$. Hence, the row-normalized version of the matrix J^T will have each column as a sparse distribution of words for each document. Recall also that the C matrix recovered by the anchor word algorithms has the dimensions $V \times K$, where each column also corresponds to a sparse distribution of words for each topic. We use a fast one-step matrix multiplication to recover topic distributions: $\theta = \bar{J}^T \times C$, where $\bar{J}^T$ is a row-normalized document-word frequency matrix, and C is the estimated coefficients of the linear combination mentioned in Equation 2.27. Unlike Gibbs sampling, which will produce a sparse vector θ_d for each document d, this fast approach will produce a dense

Corpus	Train	Test	Vocab	#Labels
20NEWS	11,250	7,485	7,318	20
CONGRESS	4,446	1,112	10,634	18
REUTERS	6,835	2,674	6,847	10

Table 5.1: Statistics of the datasets in our experiments.

vector θ_d for each document d, because each entry $\theta_{d,k}$ is a summation over all the terms:

$$\theta_{d,k} = \sum_{v}^{V} \bar{J}^{T}{}_{d,v} \times C_{v,k}.$$

Figure 5.4 shows the superior performance of the proposed approach in comparison to other approaches based on the runtime on the AMAZON dataset. We use this fast inference approach in our active learning experiments.

5.3 Quantitative Analysis of Active Learning Using Topic Models

This section describes our datasets and experiments for evaluating the effect of updating topics using supervised topic models on active learning queries and on the performance of the classifier.

5.3.1 Labeled Datasets

We use three labeled datasets: 20Newsgroups [Lang, 2007, 20NEWS], which has twenty labels; 112th U.S. congressional bills, which has eighteen labels [Poursabzi-Sangdeh et al., 2016, CONGRESS]; and Reuters [Lewis, 1997, REUTERS], with its top ten most frequent labels.

We use the frequently used train and test splits for 20NEWS and REUTERS, and we manually split the CONGRESS dataset into train and test sets. After preprocessing documents by tokenizing, removing non-alphanumeric characters, and filtering words based on TF-IDF, we perform topic modeling and active learning on the training set and compute prediction accuracy on the test set. Table 5.1 shows detailed statistics for each dataset.

5.3.2 Experiment Setup

Our first goal is to immediately measure the effect of updating topics on ALTO for active learning on multi-class classification problems. Active learning strategies deployed for this purpose are locality-based, because the selection of the next document relies only on its internal information without regard for its relationship to other documents.

We design our active learning experiments with two factors: topic model (LDA or SUPANCHOR) and topic factor in the active querying strategies (dominant topic or topic entropy). For LDA, we need to run it only once before active learning start, to get static topics. The reason we do not evaluate the ANCHOR here is because static topics produced by ANCHOR are not comparable with those produced by LDA (see Section 3.5.3). For SUPANCHOR, we run it in every active learning iteration as a new document is labeled; hence, in this case, topics are dynamically updated to reflect new label information. Thus, we simulate four conditions:

1. Active queries with updated SUPANCHOR topic features, selecting dominant topic for each document using Equation 5.2. We call this Active Learning with Supervised

Anchor Word or ALSUP.

2. Active queries with static LDA topic features, selecting dominant topic for each document using Equation 5.2 (ALTO).

3. Active queries with updated SUPANCHOR topic features, using topic entropy for each document using Equation 5.3 (ALSUP-ENT).

4. Active queries with static LDA topic features, using topic entropy for each document using Equation 5.3 (ALTO-ENT).

Following common practice in assessing the effectiveness of active learning, we iteratively query a document to label from the training set and calculate the resulting accuracy on the test set. We start with five randomly selected documents with different labels and query for the labels of 199 additional documents based on the querying strategy in each condition. The θ distribution stays fixed in the conditions that use LDA. However, for the SUPANCHOR conditions, we update θ every time a new document is labeled and use the updated θ when calculating entropy and other features for the classifier.[4]

5.3.3 Parameter Tuning

Number of Topics To choose the number of topics (K) for each dataset, we use LDA to generate topics on the train set and calculate the average topic interpretability (NPMI score) for $K \in \{10, 20, 30, 40, 50\}$. We compute the NPMI score on the top twenty topic words using 1,100,000 Wikipedia articles as the reference corpus (see detail in Section 2.3.2). We

[4] The average runtime of the SUPANCHOR for REUTERS is 33 seconds, for 20NEWS is 57 seconds, and for CONGRESS is 97 seconds.

select $K = 40$ for 20NEWS, $K = 30$ for CONGRESS, and $K = 10$ for REUTERS, as these numbers of topics generate the maximum average topic interpretability. For consistency, we use the same K for generating SUPANCHOR topics incrementally in each dataset.

SUPANCHOR Parameters Anchor word based topic models commonly use a *minimum document frequency* hyper-parameter (M) to select candidate anchor words (see Chapter 3). To set M, we run the unsupervised anchor word algorithm with fixed K and pick the $M \in \{100, 200, ..., 800\}$ that leads to the maximum held-out likelihood score on the train set. We select $M = 200$ for 20NEWS, $M = 300$ for CONGRESS, and $M = 500$ for REUTERS. We use these fixed values of M every time we update θ with SUPANCHOR.

We use logistic regression implemented in Python's Scikit-Learn library [Pedregosa et al., 2011] with topic probabilities as features to calculate entropy in the conditions that use active learning and to calculate the classifier's prediction accuracy score on the test set.[5] To account for the randomness of initial document selection, we repeat the procedure explained above five times and average the results.

5.3.4 Topic Diversity Improves Local Active Learning Strategies

ALSUP Outperforms ALTO By dynamically updating topics after each document's label becomes available, we significantly improve prediction accuracy over ALTO. For example, for the beginning iterations, in all the datasets, ALSUP significantly outperforms ALTO, except for the last iterations of the REUTERS dataset, where it seems that ALTO can catch

[5] Our preliminary examination showed that classification accuracy when the classifier uses topic features is comparable to the case when the classifier uses both topic and unigram features. Therefore, we report accuracy of classifiers using topic features only (for both LDA and SUPANCHOR).

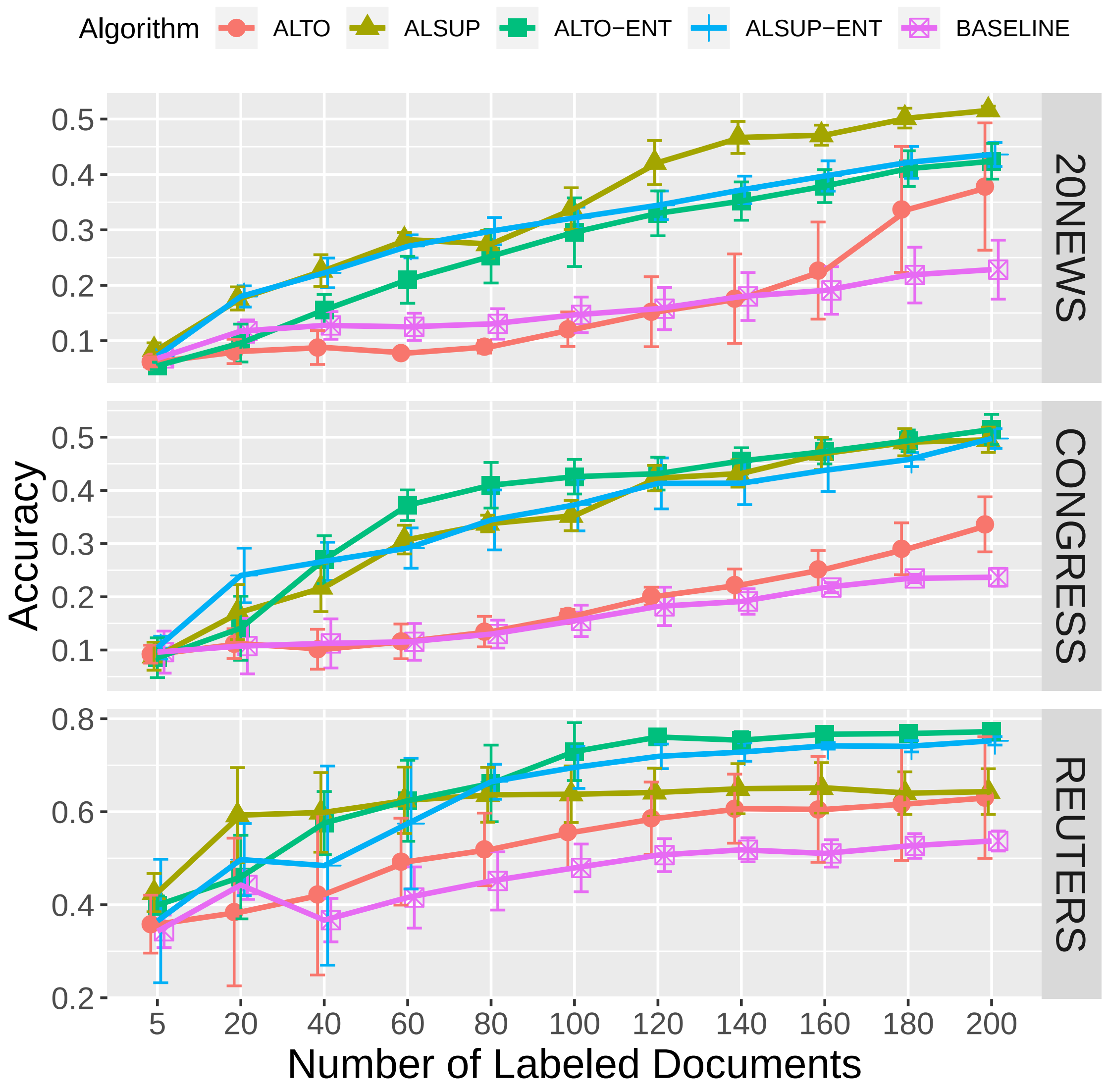


Figure 5.5: Mean classification accuracy with topic features on the test set. Error bars indicate 95% confidence intervals. Using dynamic topics generated from SUPANCHOR (ALSUP) significantly improves accuracy over using static topics generated from LDA (ALTO). Additionally, using topic entropy in active learning strategies (ALTO-ENT) is better than using the dominant topic approach (ALTO).

up with ALSUP. ALSUP is also more robust than ALTO; it fluctuates less (has a smaller standard deviation).

Topic Entropy Outperforms Dominant Topic By combining topic entropy with uncertainty sampling, we can improve the performance of ALTO. In fact, using only static topics, the new model, ALTO-ENT, is comparable to ALSUP and is much better than ALTO. We observe that ALTO's poor performance may be because using dominant topics with uncertainty sampling results in documents being presented to users that are not the best candidates—ALTO performs worse than all other strategies.

ALSUP Outperforms Baseline Active Learning We also investigated the baseline case of using pure active learning without having topic modeling. In this scenario, we use the same logistic regression classifier and uncertainty sampling strategy and train the model on the word lexical features. Figure 5.5 shows that the baseline accuracy scores are all quite low in comparison to the case of using the supervised topic model (ALSUP). The main reason for this is that, in the case of using topic models, topic features provide more compressed and meaningful views of documents; while for lexical features, training on a small number of documents captures only a sparse view of the datasets, and the classifier overfits when there are many word features but little information to learn from. As a result, the baseline classifier has low generality and produces poor performance on the test documents.

5.3.5 Escaping The Outliers: Global Active Learning Strategies with Topic Models

By relying on uncertainty sampling, active learning strategies that we deploy in both ALTO and our proposed techniques query documents based only on local information at the document level and ignore where each document is placed in relation to other documents. This type of sampling sometimes chooses outlier documents for classifiers (e.g., documents that are too short or documents that consist of infrequently used words), which often leads to worse performance [McCallum and Nigam, 1998] (see Figure 5.3).

As addressed in Section 5.1.2, the general density techniques consider global input space information to avoid selecting outliers. Because topic models have been used as a good dimensionality reduction technique and to capture semantic similarity among documents, they can be used to model the topic distribution landscape of a document collection. Combining representativeness and informativeness is not new; for example, Dasgupta and Hsu [2008] apply hierarchical clustering on unlabeled data and use learned clusters to guide active learning strategies, and Huang et al. [2014b], Du et al. [2017] combine representativeness and informativeness to design an excellent active learning strategy that works with many applications. In this section, we design four new conditions in which we combine document entropy with document representativeness to select the best document for labeling.

Using Topic Distributions for Document Similarity Measure We use topic distributions θ to measure similarity among documents from Equation 5.1:

$$\text{Sim}(d, d') = \text{Cosine\_Similarity}(\theta_d, \theta_{d'}).$$

We update each of the locality strategies with the above similarity measure. Specifically, two new strategies are

$$d^* = \arg\max_d \left(\mathbb{H}_C\left[y_d\right] \times \theta_{d,k_d} \times \left(\frac{1}{D} \sum_{i=1}^{D} \text{Sim}(d, d^{(i)}) \right)^p \right) \tag{5.4}$$

and

$$d^* = \arg\max_d \left(\mathbb{H}_C \left[y_d \right] \times \mathbb{H}_C \left[\theta_d \right] \times \left(\frac{1}{D} \sum_{i=1}^{D} \text{Sim}(d, d^{(i)}) \right)^p \right). \tag{5.5}$$

Similarly to the setup above, we design our active learning experiments with four new conditions using a representativeness measure with a fixed value of $p = 1$:

1. Active queries with updated SUPANCHOR topic features, selecting dominant topic for each document using Equation 5.4 (ALSUP-REP).

2. Active queries with static LDA topic features, selecting dominant topic for each document using Equation 5.4(ALTO-REP).

3. Active queries with updated SUPANCHOR topic features, using topic entropy for each document using Equation 5.5 (ALSUP-ENT-REP).

4. Active queries with static LDA topic features, using topic entropy for each document using Equation 5.5 (ALTO-ENT-REP).

Representativeness Helps Topic Entropy Only The active learning strategy (ALSUP-ENT-REP) that combines dynamic topics and topic entropy significantly outperforms other techniques (Figure 5.6) by adding document representativeness. One interesting observation is that adding document representativeness does not help ALTO-related strategies (Figure 5.7). One explanation for this is that, because topics are fixed in LDA, the representations of documents are not changed as more labels come in, so the relative relationship between a document and others does not change. As a result, adding this static information will not necessarily improve the selection strategy. Lastly, we see that representativeness

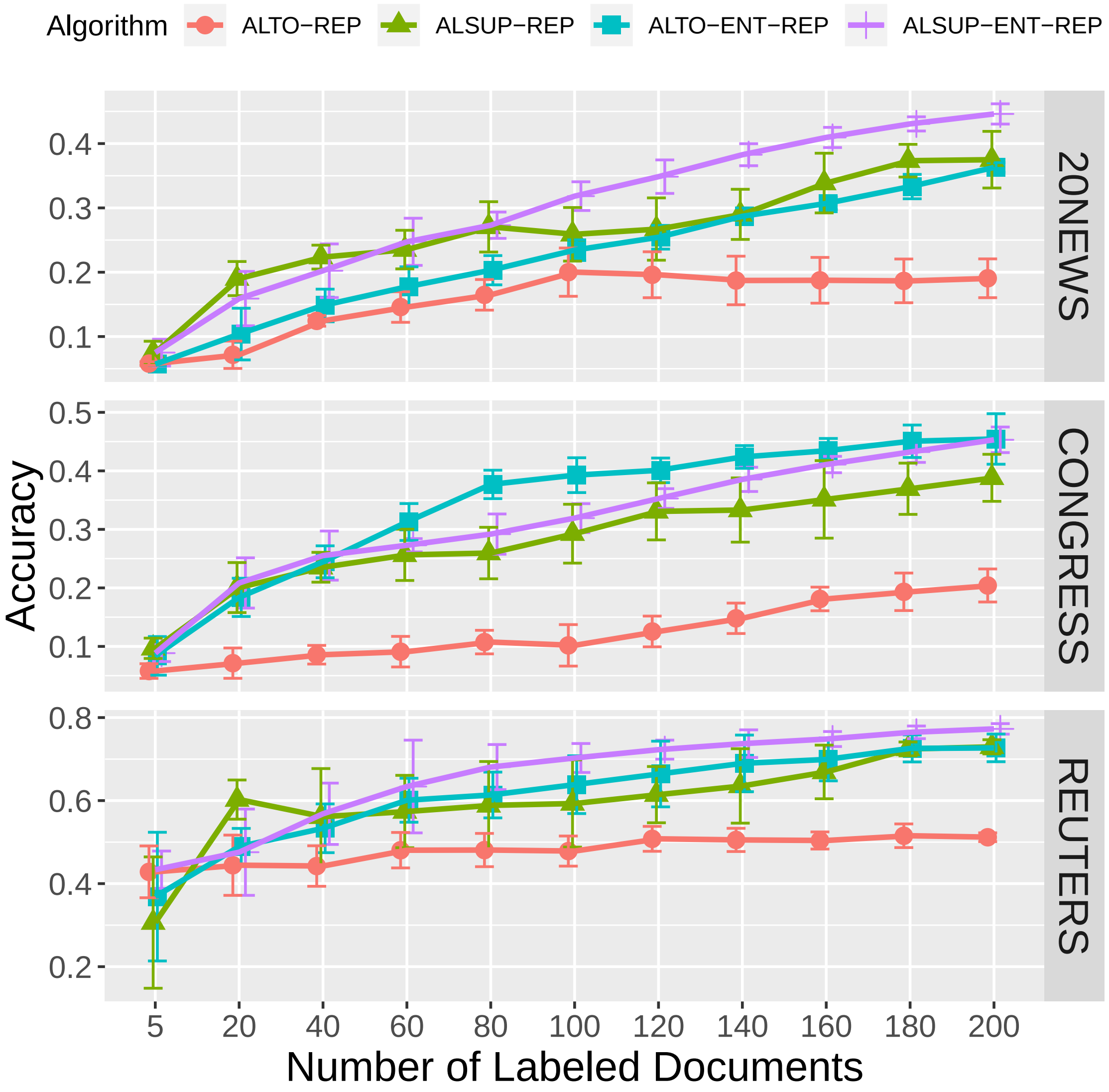


Figure 5.6: Mean classification accuracy with topic features on the test set. Error bars indicate 95% confidence intervals. SUPANCHOR using entropy and document representativeness outperforms all other techniques.

improves model robustness across all strategies, as reflected in lower standard variation values (Figure 5.7).

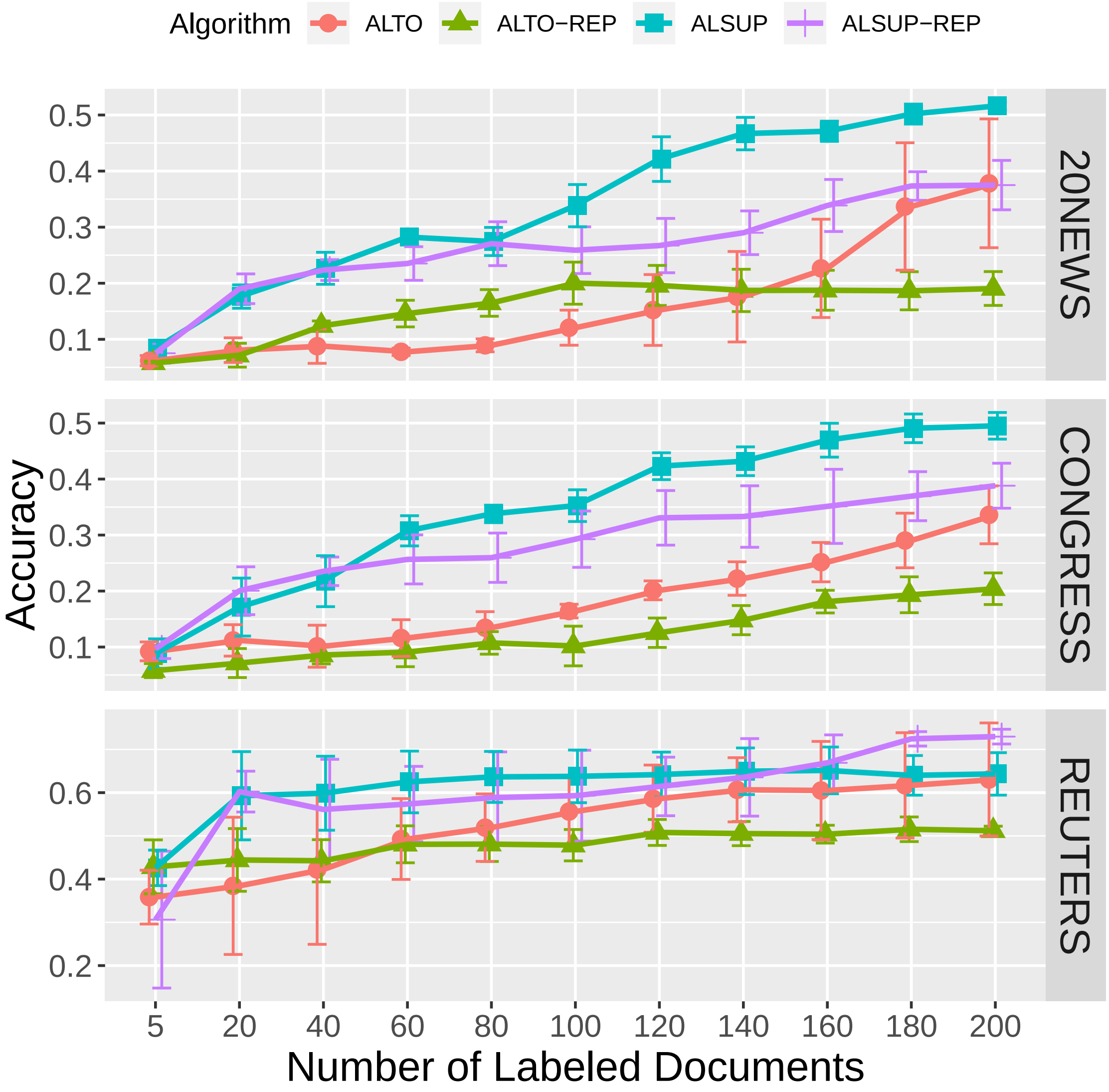


Figure 5.7: Mean classification accuracy with topic features on the test set. Error bars indicate 95% confidence intervals. Adding document representativeness does not help strategies that combine uncertainty sampling with dominant topic.

5.3.6 Topic Interpretability

In this section, we inspect the quality of topics produced by the supervised topic model as users label more documents. We use the same strategy as in Section 3.5 to compute a topic interpretability score (NPMI score as defined in Section 2.3.2) based on 1 million Wikipedia documents. Unlike ALTO, where users see only a fixed set of topics, when

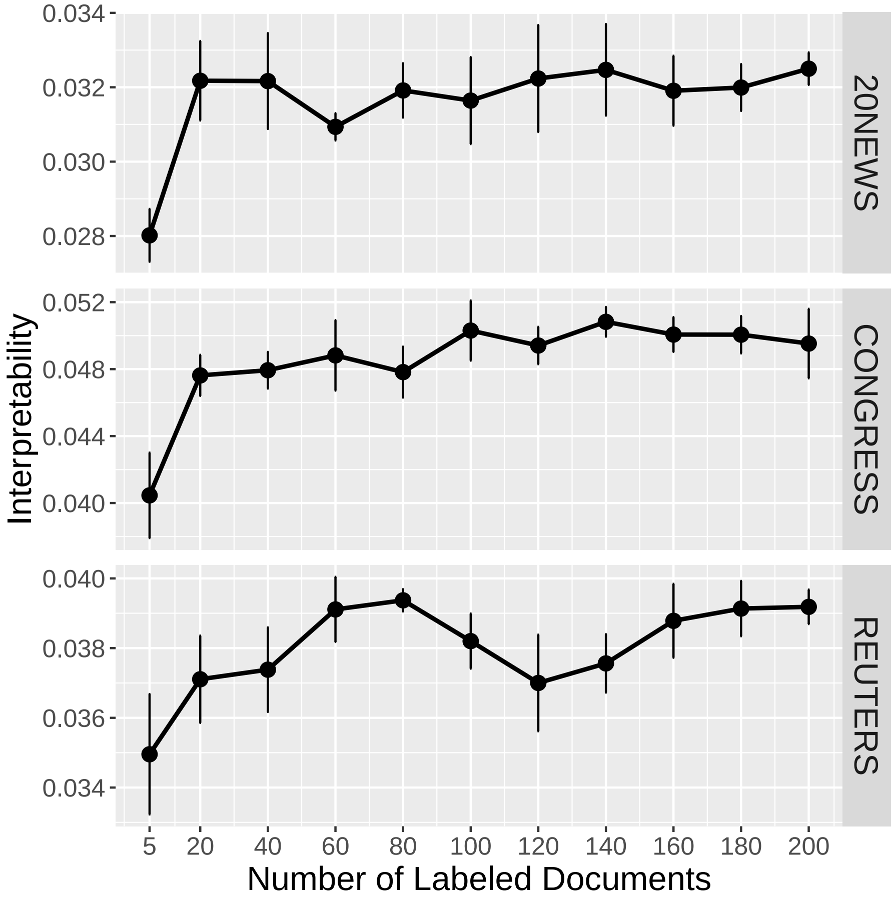
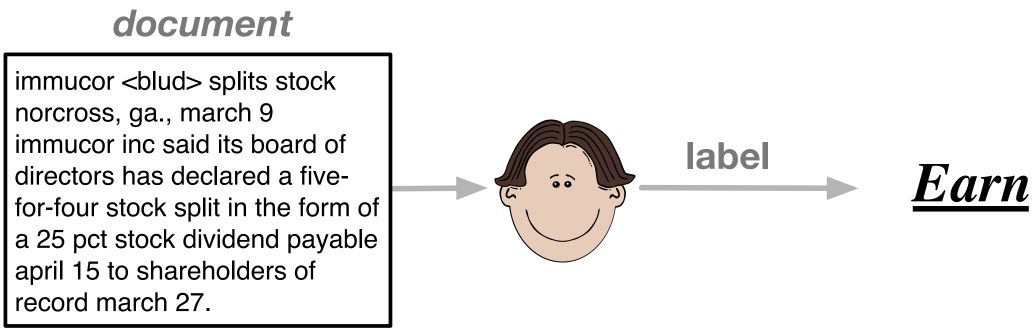


Figure 5.8: Mean topic interpretability measured as more labeled documents come into the system using SUPANCHOR. Error bars indicate 95% confidence intervals. The quality of topics improves over time as users label more documents because the supervised topic model learns new themes that capture label information.

using SUPANCHOR, users are constantly updated with new topics; these dynamic topics capture newly updated label information, hence improving topic interpretability, as seen in Figure 5.8.

Also, as we have seen in the previous chapter (Chapter 4), the topic interpretability of ANCHOR and SUPANCHOR are close in sentiment datasets (AMAZON, TRIPADVISOR, and YELP), but for the datasets in this section, it appears that we get greater improvement from SUPANCHOR compared to ANCHOR. However, these improvements are not sufficient to bridge the topic quality gap between probabilistic models (e.g., LDA using Gibbs sampling) and the ANCHOR (see Chapter 3). The average topic interpretability produced by ALSUP for the 20NEWS is 0.0335 ($min = 0.0326$ and $max = 0.0343$) compared to 0.0459 from LDA, for the CONGRESS it is 0.0511 ($min = 0.05$ and $max = 0.0523$) compared to 0.0563 from LDA, and for the REUTERS it is 0.0396 ($min = 0.0391$ and $max = 0.0401$) compared to 0.0545 from LDA. One thing to keep in mind is that topic interpretability is measured by a reference corpus (e.g., in this case Wikipedia) to capture general semantic meaning; however, showing users (annotators) specific topics that are associated with labels may be more beneficial for users, especially when ALSUP improves the topic quality overtime.

Topic Evolution with SANCHOR We investigate how topics are changed in ALSUP as users label more and more documents. In this example using the REUTERS dataset, after the active learner proposes a document, id = *earn_0002136*, for labeling, the user labels it with the label ***Earn***. Having this new document with its label, the SUPANCHOR will learn a new set of anchor words, which then recover a set of new topics; as we can see below (see also Chapter 4), only two new anchor words were introduced, and they replaced two old anchor words. However, based only on these two new anchor words, the two new topics directly reflect the influence of a newly introduced labeled document about *earning*. These new topics are more beneficial overall—although with a minor sacrifice of topic interpretability—since they are more closely related to the label (interpretability decreased from 0.0385 to 0.0345) (Figure 5.9). Eventually, after a sufficient number of labeled documents are collected, topics become more stable, and we do not see significant changes of anchor words and their associated topics.



immucor <blud> splits stock
norcross, ga., march 9
immucor inc said its board of directors has declared a five-for-four stock split in the form of a 25 pct stock dividend payable april 15 to shareholders of record march 27.

Topic Interpretability: ***0.0385 => 0.0345***

***common => stock*:** common, stock, share, group, investment, split, shares, *{international, offer, prior}*
{board, shareholders, stake}
***today => offer*:** offer, says, shares, *{effective, london, market, money, stock, today, were}*
{acquisition, group, international, merger, share, tender, unit}

— — — — — —

***bank*:** bank, this, banks, market, money, dealers, rate, *{central, dollar, national}*
{london, rates, today}
***trade*:** trade, japan, surplus, deficit, this, were, says, *{against, foreign, japanese}*
{dollar, exchange, they}
***dividend*:** dividend, financial, share, prior, june, nine, *{includes, mths split, stock}*
{franklin, group, income, seven}
***prices*:** prices, barrels, crude, price, were, production, canadian, this, reserves, *{says}*
{today}
***profit*:** profit, prior, oper, extraordinary, includes, mths, gain, shrs, nine, *{seven}*
{excludes}
***sales*:** sales, diluted, mths, prior, seven, ended, industries, shrs, nine, *{group}*
{products}

— — — — — —

***tonnes*:** tonnes, department, maize, export, corn, wheat, grain, exporters, report, usda
***quarter*:** quarter, share, first, oper, earnings, prior, sale, includes, gain, operations

Figure 5.9: Illustration of Topic Evolution: Comparing topics produced before and after a new document is labeled with the label ***Earn*** from the REUTERS dataset (top). The box (bottom) shows top ten words for each topic; anchor words are in both italic and bold. Old words (italic) are replaced by new words (bold). Two new anchor words were introduced and replaced two old anchor words (**stock** replaced **common**, and **offer** replaced **today**), producing new label-specific topics related to ***Earn***. Even though topic interpretability was decreased overall, from 0.0385 to 0.0345, due to the replacement of more general, interpretable topics (e.g., *common stock investment*) with more label-specific topics (e.g., *acquisition shareholders split*), this is actually beneficial for annotators, as they will encounter more similar documents within the scope of this ***Earn*** label.

5.3.7 Computational Complexity

Active learning frameworks utilizing local information (ALTO and ALSUP) run as fast as any normal baseline active learning frameworks using uncertainty sampling. Even in the case where we combine with topic distributions θ (Equation 5.2 and Equation 5.3), the main bottleneck is still the expense of the active learner to re-classify all unlabeled documents during each iteration. If we assume that the number of labels and the number of topics is small and finite, computing the entropy of the label distributions or the topic distributions is $O(1)$. Therefore, the computational complexity for each iteration is $O(D)$, where D is the size of the corpus. In practice, this complexity is pretty expensive for large corpora, and we should reduce it to $O(\log(D))$ using Approximate Uncertainty Sampling (AUS) [Segal et al., 2006].

Comparing to local active learning strategies, global strategies (deployed in ALTO-REP and ALSUP-REP) that use densities such as document representativeness (Equation 5.4 and Equation 5.5) are very expensive. Their computational complexity is $O(D^2)$, which is infeasible to work with large corpora. Settles and Craven [2008b] show that we need to use pre-computation or caching techniques to reduce the time required to select the next document for querying and only by doing that these strategies can work for interactive scenarios. Pre-computation can be done easily with ALTO-REP since we only compute document similarities using topic distributions from LDA once. For ALSUP-REP, however, because the SUPANCHOR updates topic distributions whenever each newly labeled document shows up, it is challenging to come up with a good pre-computation strategy to bypass the $O(D^2)$ complexity of computing similarity matrix among D documents. One solution for this problem is to use active learning in a batch mode where users only need to provide feedback in batches; in this case, ALSUP-REP needs to recompute the similarity matrix many fewer times.

5.4 Conclusions

In this chapter, we introduce the supervised anchor topic model into the active learning framework by extending the ALTO framework in Poursabzi-Sangdeh et al. [2016]. Our active learning framework, ALSUP, produces better prediction accuracy than ALTO in active learning simulations using three multi-class labeled datasets. Additionally, we introduce the concepts of topic entropy and document representativeness into ALTO, improving its performance and robustness.

A big problem in active learning is that the acquired training set built via active learning comes from a biased distribution, because it is tied to the classifier that is used to pose queries to select documents for labels. In all experiments, we use a logistic regression model due to its speed to make sure that our active learning framework satisfies the interactivity requirement. We also ignore the batch-mode in active learning, in which the active learner queries labels for a small set of documents instead of for only one document at a time. One interesting direction would be to combine more accurate supervised models with batch-mode while still satisfying performance requirements.

Accurate document classification relies on the quality of the labeled training set. Active learning strategies such as uncertainty sampling have been shown to produce a high-quality training set with minimal effort. Many other authors, however, have published negative results regarding active learning strategies. For example Guo and Schuurmans [2007] show that some query strategies are much worse than random sampling, and Gasperin [2009] report negative results for active learning in an anaphora resolution task. In some cases, researchers find inconsistency in how well active learning performance correlates with the proficiency of the annotator [Baldridge and Palmer, 2009]. In a comprehensive survey of the active learning landscape, Settles [2010] report statistics showing 91% of researchers who use active learning in annotation projects find it satisfies their needs, but about 20% of them opt out of using active learning for various reasons. One immediate direction for future research is to extend our framework beyond uncertainty sampling strategies. Furthermore, we believe these methods should eventually be empirically evaluated in interactive frameworks with human annotators to ensure effectiveness in real-world scenarios.

Finally, the evolution of topics as users label documents is particularly interesting in the sense that we can get feedback from users not only for labels but also for the topics. A simple way to do this would be to allow users to pick “good” words in a topic and give those a stronger prior for that topic when iteratively updating the model [Musialek et al., 2016]. We believe this research direction will provide a powerful framework for active learning to tackle the challenging problem of label collection as well as label understanding through the iterative process of interacting with supervised topic models.

Chapter 6

Conclusion and Future Work

Capturing semantic meaning from extensive collections of text documents is a very challenging task. Probabilistic Bayesian approaches to this problem have been successful in working with small to midsize datasets but suffer when working with large datasets; they often run slowly, hence creating a high-latency system. In this thesis, we present several new topic models based on the anchor word algorithm by Arora et al. [2013]: the regularized anchor word topic models and the supervised anchor word topic model. We also introduce a new active learning framework that combines the supervised anchor word topic model with uncertainty sampling. Our results contribute to the current literature of analytical models that support working with enormous datasets for scalability, variability, and interactivity. This final chapter summarizes the contributions of these approaches and discusses some directions for future work.

6.1 Summary of Technical Contributions

In Chapter 3, we present the *regularized anchor word algorithms* that improve upon the original anchor word algorithm (ANCHOR) to provide (1) a more flexible and extensible model for handling external knowledge and (2) a model with higher topic quality. The new models take advantage of the usefulness of regularization and priors in machine learning to build more robust and stable algorithms. We formulate the Gaussian prior as an

L_2-regularization term and enhance the ANCHOR model to create the ANCHOR-L_2 model. As a result of this enhancement, the ANCHOR-L_2 is flexible, since we can easily inject informed priors (external knowledge) into the model just by changing the mean of the appropriate Gaussian distribution. We show a successful application of the ANCHOR-L_2 for incorporating Linguistic Inquiry and Word Count [Pennebaker and Francis, 1999, LIWC] knowledge about word association into a topic model. In addition, we create the ANCHOR-BETA to improve the quality of topics by adding a $Beta$ regularization term to the objective function of the anchor word algorithm. We formulate the $Beta$ regularizer from the formulation of a Dirichlet distribution, a popular distribution for many probabilistic topic models. A new contribution in this chapter is a thorough evaluation of the anchor word algorithms against Gibbs sampling and variational inference using held-out likelihood and topic interpretability. Using three datasets—20 Newsgroups (20NEWS), Neural Information Processing Systems articles (NIPS), and New York Times articles (NYT)—we show through extensive experiments that the regularized anchor word topic models are very effective in terms of both model capacity and topic quality.

In Chapter 4, we introduce the supervised anchor word algorithm for prediction and for gaining insight into labeled datasets. The original anchor word algorithm represents a word as a vector of the conditional probabilities of all other words given the observation of that word (word-word co-occurrence statistics); the supervised anchor word algorithm enriches this representation with the conditional probabilities of all labels given the observation of that word (word-label co-occurrence statistics). Using three sentiment datasets—Amazon product reviews, TripAdvisor hotel reviews, and Yelp restaurant reviews—we show through experiments the superior performance of the supervised anchor word algorithm against the anchor word algorithm and the SLDA, in terms of both speed and prediction accuracy. We also show the advantage of using the supervised anchor word algorithm for prediction on sentiment datasets: it learns a new set of anchor words that have a strong association with the sentiment levels, and these anchor words help explain more about the sentiment datasets—hence, providing more insights for users.

In Chapter 5, we go beyond the standard utility of supervised topic models for prediction and insights by presenting a new application of the supervised anchor word topic model for active learning. We introduce ALSUP, an active learning framework that combines uncertainty sampling with the dynamic topic representation of documents produced by the supervised anchor word topic model. Quantitative experiments on predicting multiple labels using three labeled datasets—the Twenty NewsGroups, the 112th U.S. congressional bills, and the Reuters corpus—show that ALSUP outperforms both ALTO and a common baseline active learning model. In addition, we show the advantages of ALSUP over traditional uncertainty sampling and the ALTO model for label annotation. First, ALSUP can quickly learn new topic models that capture updated label information interactively, providing users with the latest overview of the corpus. Second, ALSUP produces an additional layer of document representation through dynamic topics that supplement uncertainty sampling for new and effective active learning strategies.

6.2 Future Work

Human Evaluation for Active Learning Using Supervised Anchor Word Chapter 5 introduces a novel active learning framework that combines supervised topic models and uncertainty sampling. Our evaluation, however, focuses on measuring the accuracy of the learned classifier on the set of labeled documents; in this case, we must assume that the set of labels is fixed, and labels provided by annotators are accurate. As addressed by ALTO, in practice, not all labels are given, and many tasks require users to come up with useful labels (e.g., in coding survey responses). Given the benefits of preparing the label set for active learning, we believe that dynamic topics in a framework such as ALSUP will provide many advantages compared to traditional active learning. For example, we can extend ALSUP to (1) inform annotators of new topics and (2) accept feedback from annotators to refine topics. This approach may be very beneficial for annotation in cost-sensitive active learning [Haertel et al., 2008b, Zaidan et al., 2008, Donmez and Carbonell, 2008, Ringger et al., 2008, Haertel et al., 2008a], reducing costs associated with labeling efforts and loss associated with mislabeling.

Improving Topic Quality of Supervised Anchor Word Even though the supervised anchor word algorithm produces superior accuracy and serves well for prediction and insights, improving topic quality of supervised topic models is not the focus of this thesis. We focus mainly on capturing the label information into the anchor words and using these anchor words to reconstruct insightful topics. However, Section 4.5.2 shows that the topic quality of the supervised anchor word is rarely better than the topic quality of the original anchor word algorithm. We believe that developing a supervised topic model that produces insightful and high-quality topics is challenging, yet would be particularly beneficial, especially in guiding the development of label sets for active learning. One approach to addressing this challenge is to use the $Beta$ regularization technique used in the Chapter 3, but speed is a significant issue because the ANCHOR-BETA model often runs at only one-tenth the speed of the SUP-ANCHOR, thus, combining $Beta$ regularization with the supervised anchor word topic model may work in settings that do not involve humans (e.g., active learning). Recent anchor word models [Lee et al., 2015, Lee and Mimno, 2014b, Lund et al., 2017] introduce new techniques to improve the topic quality of the ANCHOR without incurring more computation costs. Combining the supervised anchor word model with one of these techniques is worth considering in the future.

Connection with Deep Learning Deep learning [LeCun et al., 2015] has become a powerful workhorse for machine learning research in recent years. Deep learning comprises computational models that utilize multiple processing layers that learn representations of data at various levels of abstraction. Applications of deep learning to natural language processing have produced state-of-the-art models for speech recognition [Hinton et al., 2012, Graves et al., 2013], machine translation [Cho et al., 2014, Bahdanau et al., 2014, Luong et al., 2015], and language modeling [Collobert and Weston, 2008]. Success stories of deep learning are also associated with word embedding models such as Word2Vec [Mikolov et al., 2013b] or Glove [Pennington et al., 2014], since these models successfully capture interesting language properties for tackling problems related to word analogies [Baroni et al., 2014] or question answering [Iyyer et al., 2014]. There are many commonalities between techniques that learn word representations using deep learning and those that learn word representations using matrix factorization. For example, Levy and Goldberg [2014b] show that the skip-gram with negative-sampling deployed in the Word2Vec model is equivalent to factorizing a word-context matrix where cell values are the pointwise mutual information of the respective word and context. We can also see the similarity between these embedding techniques and the anchor word algorithms. The anchor word algorithms use nonnegative matrix factorization to factorize word and word matrix to recover topic models, and the resulting coefficient vector $\boldsymbol{C}_i$ for a word index i is a word embedding in a K (the number of topics) dimensional space.

Although all are motivated by specific problems related to practical applications such as topic quality, insights, and prediction, many models presented in this thesis can be applied in conjunction with deep learning models. One direction to improve the quality of topics produced by anchor word models is to combine word representation using co-occurrence statistics with pre-trained word embeddings using deep models for high interpretability. The advantage of this approach is the ability to embed knowledge from different domains into the anchor word topic models. For example, word embeddings learned using PubMed [Pyysalo et al., 2013] can ensure that more medical terms appear in anchor word topics for greater interpretability in medical contexts.

Topic Inference for Anchor Word Algorithms Thus far, all the anchor word algorithms learn topic models by recovering the topic word distributions (matrix A), but they cannot recover the document topic distributions (matrix θ). These document topic distributions are essential for downstream applications (e.g., classification), and we recover them by approximate methods using either Gibbs sampling or variational inference (see Chapter 3 and Chapter 4). However, using probabilistic inference methods is slow, and it produces only approximate solutions [Yao et al., 2009a, Lee et al., 2017b]. In addition, these probabilistic inference schemes require specifying a prior parameter of the document topic distributions (e.g., α in a Dirichlet distribution), and estimating topics with a Dirichlet prior is challenging [Sontag and Roy, 2011]. Using direct multiplication (see Section 5.2.2 and also Lee et al. [2017b]) is fast, and it works very well for prediction problems. However, document topic distributions are not sparse, so these outputs cannot be used for tasks that require interpretable results for document-topic relationships. We believe that accurately inferring these quantities is very challenging yet highly useful, especially in helping supervised prediction tasks and for the development of future models.

To the best of our knowledge, there are only two works that try to address this problem. The first is the Thresholded Linear Inverse (TLI) model proposed by Arora et al. [2016], and the second is the Prior-Aware Dual Decomposition (PADD) model by Lee et al. [2017b]. Even though PADD shows better performance than TLI using both synthetic documents and real documents, PADD has to solve a large number of nonlinear optimization problems and requires heavy tuning of the learning rate. Both models also use matrix inversion, which is often unstable, and they are not fast enough for interactive settings.

One exciting direction is to directly recover the document topic vector for each document after the recovery of the matrix A by forming this document-topic inference problem as a linear regression problem. Specifically, representing a document d as a vector of size V of word frequencies y_d, this empirical distribution of words in a document d is an approximation of $A \times x_d$: $y_d = A \times x_d + noise$. We can transform it using logarithmic Chebyshev approximation with L_∞ norm [Vandenberghe and Boyd, 1996]. The newly formed problem can be solved either with gradient methods or with a semidefinite programming solution. Unlike Arora et al. [2016] or Lee et al. [2017b], this approach may not need to impose any assumption on the topic matrix A (e.g., condition numbers), and it can scale well.

Extending the Applications of Anchor Word Algorithms Learning word representations from text and using them to support understanding is a never-ending quest for NLP research. Scalable topic models such as anchor word models play an important role in this quest. Compared to probabilistic topic models, anchor word models operate only on word-word co-occurrence statistics; this brings many advantages, such as a deterministic training step or a provable guarantee given the existence of anchor words. Many new anchor word models have been introduced in recent years to address many practical issues. Typically used for topic quality are the regularized anchor word models in Chapter 3, the Tandem multiword anchor models in Lund et al. [2017], the rectified anchor word model in Lee et al. [2015], or the low-dimensional anchor word model by Lee and Mimno [2014b]; for topic hierarchy is the ADMM-DR by Lee et al. [2017a]; and for document-topic inference are the LTI model [Arora et al., 2016] or the PADD model [Lee et al., 2017b]. However, compared to probabilistic topic models, spectral topic models generally, and anchor word topic models in particular, are still limited in terms of applications and adoption; most users still turn to sampling-based models for convenience. One solution is to increase the variety of anchor word topic models available for more practical situations; scarcity of appropriate solutions is not a problem we see with probabilistic topic models, given their long history.

The advantages of anchor word methods for interactive settings, accurate prediction, and insights are essential for many real-world applications. Communicating information about anchor word methods to users is much easier than communicating information about probabilistic models, because the outputs of anchor word algorithms are deterministic and more intuitive. Disseminating anchor word methods to more application domains requires collaborations with scientists from those domains.

The strong assumption that anchor words exist in most datasets is verified in theory [Ding et al., 2015]; it is equally important to show this property in practical domain-specific datasets. One effective way to do this is through visualization of anchor words and their associated topics. We believe that anchor word methods will help researchers and users discover more useful results through visualization and interactive feedback from domain experts.

Throughout this thesis, we have shown the efficiency of the anchor word algorithms in comparison to other non-anchor algorithms for topic modeling. The original benchmark on the synthetic NIPS documents by Arora et al. [2013] (Figure 2.7) and our runtime analysis (Section 3.5.4) show that the original anchor word algorithm (ANCHOR), the regularized anchor word algorithms (ANCHOR-L_2 and ANCHOR-BETA) achieve a roughly constant training time (as opposed to Gibbs sampling or variational inference algorithms which are linear to the size of the corpus). Because of this, these anchor word algorithms can scale up to truly huge document collections, e.g., hundreds of millions of documents. We limit our evaluations using smaller datasets, where the number of documents is in the range of thousands (from around a thousand NIPS documents to around a hundred thousand TRIPADVISOR documents), mainly due to the inability of non-anchor models to handle large datasets.

The ALSUP model in an interactive user case, however, has its limitation. We show in the complexity analysis in the Section 5.3.7 that ALSUP has its complexity of $O(D)$ in each active learning iteration; this includes training SUPANCHOR, estimation of document topic matrix θ, re-classifying unlabeled documents, and computing entropies. Since the training time of SUPANCHOR is a constant, as the number of documents grows (around one hundred thousand documents), the three later operations, which each has a complexity of $O(D)$, surpass SUPANCHOR training time. A quick test using Numpy [Van Der Walt et al., 2011] and Scipy [Jones et al., 2001] python packages to create a synthetic dataset for a vocabulary size of $V = 1000$ and a number of topics $K = 20$ shows that three later operations only take about 1 second for a corpus size of one million documents but take a couple of minutes for tens of millions of documents. Experience using interactive topic modeling in a commercial setting suggests that for large datasets users are content to do a large batch of feedback and then wait up to 20 minutes or more for a recalculation of the model, as long as the total number of iterations is relatively small (Philip Resnik, personal communication). In this setting, ALSUP can handle a corpus size of roughly tens of millions of documents. Similarly, if users are content doing more iterations but willing to accept only 60 seconds per recalculation, then ALSUP can handle a corpus size of roughly a million documents.

6.3 Limitations

One limitation of the anchor word algorithms is the space required to store the Q matrix; as the vocabulary grows, this matrix becomes extremely large. A more compact representation for words may (1) learn a new set of anchor words more quickly and accurately and (2) recover the model parameters more quickly. A possible way to address this is to use word embeddings in lower dimensions for Q. Another approach is to use a distributed framework for anchor word algorithms because it is simple to parallelize them.

Anchor word algorithms run much faster than Gibbs sampling or variational inference. However, there is a brief window in which this starts to happen. Anchor word algorithms take $O(D)$ to estimate a co-occurrence matrix Q only once, while Gibbs sampling or variational inference must pass several hundred times (or even thousands of times) through the corpus. Two additional steps of the anchor word algorithms are learning anchor words (which take $O(V^2)$) and recovering topic coefficients (which take $O(K \times V^2)$). Asymptotically, as the number of documents (D) becomes very large compared to the number of word types (V) and the number of topics (K), we observe significantly more effective performance by the anchor word algorithms compared to sampling techniques. In the case of very large datasets, the running time of these anchor word algorithms is effectively independent of the size of the corpus. For smaller datasets where V is close to D, without parallel implementations, anchor word algorithms can actually run slower than Gibbs sampling (see Figure 2.7).

For prediction tasks, complex models, such as ensemble or deep learning models, produce state-of-the-art accuracy on most large labeled datasets. However, these models often lack the interpretability that would allow us to explain why they perform so effectively. In response, various methods have recently been proposed to help resolve this tension between *accurary* and *interpretability* [Bach et al., 2015, Ribeiro et al., 2016, Chen et al., 2016, Lundberg and Lee, 2017]. Our supervised anchor word model produces both high accuracy and insights, blending the power of understanding both unlabeled data through topic models and labeled data through anchor words. For future work, it would be worthwhile to evaluate the performance of supervised anchor words with these complex models for text classification.

One weakness of the supervised anchor word algorithm is that the prediction step must still require a classifier (such as logistic regression or SVM). A potential future direction for research would be to embed the prediction step internally within the model, such as in the SLDA. Having direct prediction would allow wider application of the supervised anchor word model.

Finally, active learning using supervised topic models is a novel idea and is worthy of further investigation beyond the supervised anchor word algorithm. Stochastic inference [Hoffman et al., 2013] and online learning [Hoffman et al., 2010, Zhai and Boyd-Graber, 2013] have substantially increased the speed of probabilistic models, although effectively implementing these models is not trivial, and the latency is still high compared to the anchor word algorithms. We believe that comparing these two fields of research in the setting of active learning will be valuable for future work.

K	number of topics
V	vocabulary size
Q	word co-occurrence matrix $Q_{i,j} = p(w_1 = i, w_2 = j)$
$\bar{Q}$	conditional distribution of Q $\bar{Q}_{i,j} = p(w_1 = j \,\vert\, w_2 = i)$
$\bar{\boldsymbol{Q}}_{i,\cdot}$	row i of $\bar{Q}$
A	topic matrix, of size $V \times K$ $A_{j,k} = p(w = j \,\vert\, z = k)$
C	anchor coefficient of size $V \times K$ $C_{j,k} = p(z = k \,\vert\, w = j)$
$\mathcal{G}$	set of anchor word indexes $\{g_1, \dots g_K\}$
ξ	regularization weight
$[U]_i$	the i^{th} element of a vector U
J^T	the transpose matrix of a matrix J

Table A.1: Notation used. Vectors are in bold ($\bar{\boldsymbol{Q}}_{i,\cdot}, \boldsymbol{C}_{i,\cdot}$), sets are in script $\mathcal{G}$

Appendix A

Derivations for The Anchor Word Algorithms

A.1 Notations

Given a vector **x** $= (x_1, x_2, \dots, x_n)$ of **R**^{n}, L_2 norm is defined as:

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

The probability density function of the beta distribution with two parameters $a > 0$ and $b > 0$, for $0 \leq x \leq 1$ is defined as

$$Beta(x; a, b) = \frac{x^{a-1}(1-x)^{b-1}}{\boldsymbol{B}(a, b)},$$

where $\boldsymbol{B}(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ and $\Gamma(z) \doteq \int x^{z-1}e^{-x}dx$ is the gamma function.

The objective function for the ANCHOR algorithm is

$$\boldsymbol{C}_{i,\cdot} = \text{argmin}_{\boldsymbol{C}_{i,\cdot}} D_{\text{KL}} \left(\bar{\boldsymbol{Q}}_{i,\cdot} \,||\, \sum_{g_k \in \mathcal{G}} C_{i,k} \bar{\boldsymbol{Q}}_{g_k,\cdot} \right). \tag{A.1}$$

Recovering topic matrix A requires us to find solution for $\boldsymbol{C}_{i,\cdot}$ for each word i from the vocabulary; this step is parallelable. Next, we present different priors on the matrix C by applying different regularizations.

A.2 ANCHOR-L_2 Regularizer

We apply L_2 regularization which is equivalent to a Gaussian prior. Adding the L_2 regularizer to the Equation A.1 results in the following objective function:

$$\begin{aligned} J_{gauss} &\equiv D_{\text{KL}} \left(\bar{\boldsymbol{Q}}_{i,\cdot} \,||\, \sum_{g_k \in G} C_{i,k} \bar{\boldsymbol{Q}}_{g_k,\cdot} \right) + \xi \|C_{i,\cdot} - \mu_{i,\cdot}\|_2 \\ &= \sum_{v=1}^{V} \bar{Q}_{i,v} (\log(\bar{Q}_{i,v}) - \log(\sum_{g_k \in \mathcal{G}} C_{i,k} \bar{Q}_{g_k,v})) + \xi \|\boldsymbol{C}_{i,\cdot} - \boldsymbol{\mu}_{i,\cdot}\|_2. \end{aligned} \tag{A.2}$$

Taking derivative of J_{gauss} with regards to $C_{i,k}$, we have

$$\frac{\partial J_{gauss}}{\partial C_{i,k}} = -\sum_{v=1}^{V} \bar{Q}_{g_k,v} \frac{\bar{Q}_{i,v}}{\sum_{g_k \in \mathcal{G}} C_{i,k} \bar{Q}_{g_k,v}} + \xi \frac{C_{i,k} - \mu_{i,k}}{\|\boldsymbol{C}_{i,\cdot} - \boldsymbol{\mu}_{i,\cdot}\|_2}.$$

Or

$$\frac{\partial J_{gauss}}{\partial C_{i,k}} = \left[- \left[\frac{\bar{\boldsymbol{Q}}_{i,\cdot}}{\boldsymbol{C}_{i,\cdot} \bar{Q}_{\mathcal{G}}} \right] [\bar{Q}_{\mathcal{G}}]^T + \xi \frac{\boldsymbol{C}_{i,\cdot} - \boldsymbol{\mu}_{i,\cdot}}{\|\boldsymbol{C}_{i,\cdot} - \boldsymbol{\mu}_{i,\cdot}\|_2} \right]_k, \forall g_k \in \mathcal{G}, \tag{A.3}$$

where the matrix $\bar{Q}_{\mathcal{G}}$ (of size $K \times V$) is the submatrix of $\bar{Q}$ corresponding the row indexes belonging $\mathcal{G}$. The Equation A.3 shows that we can compute the gradient of J_{gauss} by first computing the matrix in the bracket.

A.3 ANCHOR-BETA Regularizer

We apply $Beta$ regularization, which is equivalent to assuming a Dirichlet prior on the topic word distribution. Adding $Beta$ regularizer to the Equation A.1 produces the new objective function:

$$J_{dir} \equiv D_{\text{KL}}\left(\bar{\boldsymbol{Q}}_{i,\cdot} \mid\mid \sum_{g_k \in \mathcal{G}} C_{i,k} \bar{\boldsymbol{Q}}_{g_k,\cdot}\right) - \xi \sum_{g_k \in \mathcal{G}} \log(\text{Beta}(A_{i,k}; a, b)) \tag{A.4}$$

$$= J_1 - \xi J_2, \tag{A.5}$$

where J_1 is the objective function of the ANCHOR, and J_2 can be written as

$$J_2 = \sum_{g_k \in \mathcal{G}} (a-1)\log(A_{i,k}) + (b-1)\log(1 - A_{i,k}) - const. \tag{A.6}$$

The topic matrix A is a linear transformation of the coefficient matrix C (Equation 2.28). Set $T_i = \sum_{v=1}^{V} Q_{i,v}$, we have

$$A_{i,k} = \frac{T_i C_{i,k}}{\sum_{v=1}^{V} T_v C_{v,k}}.$$

Define $T = [T_1, T_2, \ldots, T_V]$, then $\sum_{v=1}^{V} T_v C_{v,k} = [TC]_k$, and $A_{i,k} = \frac{T_i C_{i,k}}{[TC]_k}$. Substitute into Equation A.6, we have

$$\begin{aligned} J_2 &= \sum_{g_k \in \mathcal{G}} \{(a-1)(\log(T_i C_{i,k}) - \log([TC]_k)) + (b-1)\log(1 - \frac{T_i C_{i,k}}{[TC]_k})\} \\ &= \sum_{g_k \in \mathcal{G}} \{(a-1)\log(T_i C_{i,k}) + (b-1)\log([TC]_k - T_i C_{i,k}) + (2-a-b)\log([TC]_k)\}. \end{aligned}$$

Taking derivative for a given $C_{i,k}$, we have

$$\frac{\partial J_2}{\partial C_{i,k}} = \left[\frac{a-1}{\boldsymbol{C}_{i,\cdot}} + T_i \frac{2-a-b}{TC}\right]_k, \forall g_k \in \mathcal{G}.$$

Together with J_1 from Equation A.3, we get

$$\frac{\partial J_{dir}}{\partial C_{i,k}} = \left[-\left[\frac{\bar{\boldsymbol{Q}}_{i,\cdot}}{\boldsymbol{C}_{i,\cdot}\bar{Q}_{\mathcal{G}}}\right][\bar{Q}_{\mathcal{G}}]^T - \xi\left(\frac{a-1}{\boldsymbol{C}_{i,\cdot}} + T_i\frac{2-a-b}{TC}\right)\right]_k, \forall g_k \in \mathcal{G}. \tag{A.7}$$

Since we solve each $\boldsymbol{C}_{i,\cdot}$ independently, we can fix the matrix C from previous result batch and update the gradient as in Equation A.7 (Algorithm 3).

A.4 Proof the Linear Combination of the Supervised Anchor Word

We use the proof for the linear combination for the anchor word algorithm from Arora et al. [2013]. Using the notations from Table A.1, for any anchor word g_k, we have:

$$\begin{aligned}
\bar{Q}_{g_k,j} &\equiv p(w_2 = j \,|\, w_1 = g_k) \\
&= \sum_{k'} p(z_1 = k' \,|\, w_1 = g_k) p(w_2 = j \,|\, z_1 = k') \qquad \text{(A.8)} \\
&= p(w_2 = j \,|\, z_1 = k), \qquad \text{(A.9)}
\end{aligned}$$

where A.8 uses the assumption of admixture model where w_1 and w_2 are independent given z_1, or $w_2 \perp w_1 \,|\, z_1$, and A.9 is because $p(z_1 = k \,|\, w_1 = g_k) = 1$ and $p(z_1 = k' \,|\, w_1 = g_k) = 0, k' \neq k$ due to the anchor word assumption.

Recall from Section 4.2.1 that the matrix S of augmented representation is formed by concatenating the matrix $\bar{Q}$ of size $V \times V$ of word-word co-occurrences with a matrix of size $V \times L$ of word-label co-occurrences, where L is the number of labels (or sentiment levels): $S_{i,V+l} \equiv p(y = y^{(l)} \,|\, w = i)$.

Given an anchor word $g_k \in \mathcal{G}$, for any $j \leq V$: $S_{g_k,j} \equiv \bar{Q}_{g_k,j}$. For any $j > V$, we can represent $j = V + l$ where $1 < l \leq L$. We use the same analysis as above, assuming also the admixture property for label and topic assignment holds, $y \perp w \,|\, z$, we have:

$$\begin{aligned} S_{g_k,j} &\equiv S_{g_k,V+l} \\ &\equiv p(y = y^{(l)} \,|\, w = g_k) \\ &= \sum_{k'} p(z = k' \,|\, w = g_k) p(y = y^{(l)} \,|\, z = k') \qquad \text{(A.10)} \\ &= p(y = y^{(l)} \,|\, z = k). \qquad \text{(A.11)} \end{aligned}$$

For any other word i and $j \leq V$, we use the linear combination from the ANCHOR [Arora et al., 2013]:

$$S_{i,j} \equiv \bar{Q}_{i,j} = \sum_{k} p(z_1 = k \,|\, w_1 = i) p(w_2 = j \,|\, z_1 = k). \qquad \text{(A.12)}$$

Because $C_{i,k} = p(z = k \,|\, w = i)$ is the conditional probability of observing topic k given seeing the word i in the document, combine the Equation A.12 with the Equation A.9, we have $S_{i,j} = \sum_k C_{i,k} S_{g_k,j}$. Similarly, for any other word i and j larger than V, $j = V + l$, we apply the Equation A.11 to have to following formulation

$$
\begin{aligned}
S_{i,j} &\equiv S_{i,V+l} \\
&\equiv p(y = y^{(l)} \mid w = i) \\
&= \sum_k p(z = k \mid w = i) p(y = y^{(l)} \mid z = k) \\
&= \sum_k C_{i,k} S_{g_k,V+l} \\
&= \sum_k C_{i,k} S_{g_k,j}.
\end{aligned}
$$

Hence, for all i, j, $S_{i,j} = \sum_k C_{i,k} S_{g_k,j}$. This shows that any row of S lies in the convex hull of the rows corresponding to the anchor words, preserving the linear combination property of the unsupervised anchor word algorithm.

A.5 Finding Anchor Words with FastAnchorWords

For completeness, we describe the FastAnchorWords algorithm [Arora et al., 2013] to find anchor words for the anchor word algorithms. In this algorithm, span($\mathcal{G}$) denotes the subspace spanned by the points in the set $\mathcal{G}$. To find the farthest point, we need to compute the distance from a point x to the subspace span($\mathcal{G}$) by computing the norm of the projection of x onto the orthogonal complement of span($\mathcal{G}$).

Algorithm 5 FastAnchorWords [Arora et al., 2013]

Input: V points $\{d_1, d_2, \ldots, d_V\}$ in V dimensions, almost in a simplex with K vertices and $\epsilon > 0$.
Output: K points that are close to the vertices of the simplex.

```
Project the points $d_i$ to a randomly chosen $4\log\frac{V}{\epsilon^2}$ dimensional subspace $\mathcal{G} \leftarrow d_i$ such that $d_i$
is the farthest point from the origin.
for $i = 1$ TO $K - 1$ do
    Let $d_j$ be the point in $\{d_1, \ldots, d_V\}$ that has the largest distance to span($\mathcal{G}$).
    $\mathcal{G} \leftarrow \mathcal{G} \cup \{d_j\}$
$\mathcal{G} = \{v_1^{'}, v_2^{'}, \ldots, v_K^{'}\}$.
for $i = 1$ TO $K$ do
    Let $d_j$ be the point that has the largest distance to span($\{v_1^{'}, v_2^{'}, \ldots, v_K^{'}\} \setminus \{v_i^{'}\}$).
    Update $v_i^{'}$ to $d_j$
return $\{v_1^{'}, v_2^{'}, \ldots, v_K^{'}\}$
```

Appendix B

Additional Results for the Regularized Anchor Algorithms

B.1 Heldout Likelihood Score using $Beta$ regularization

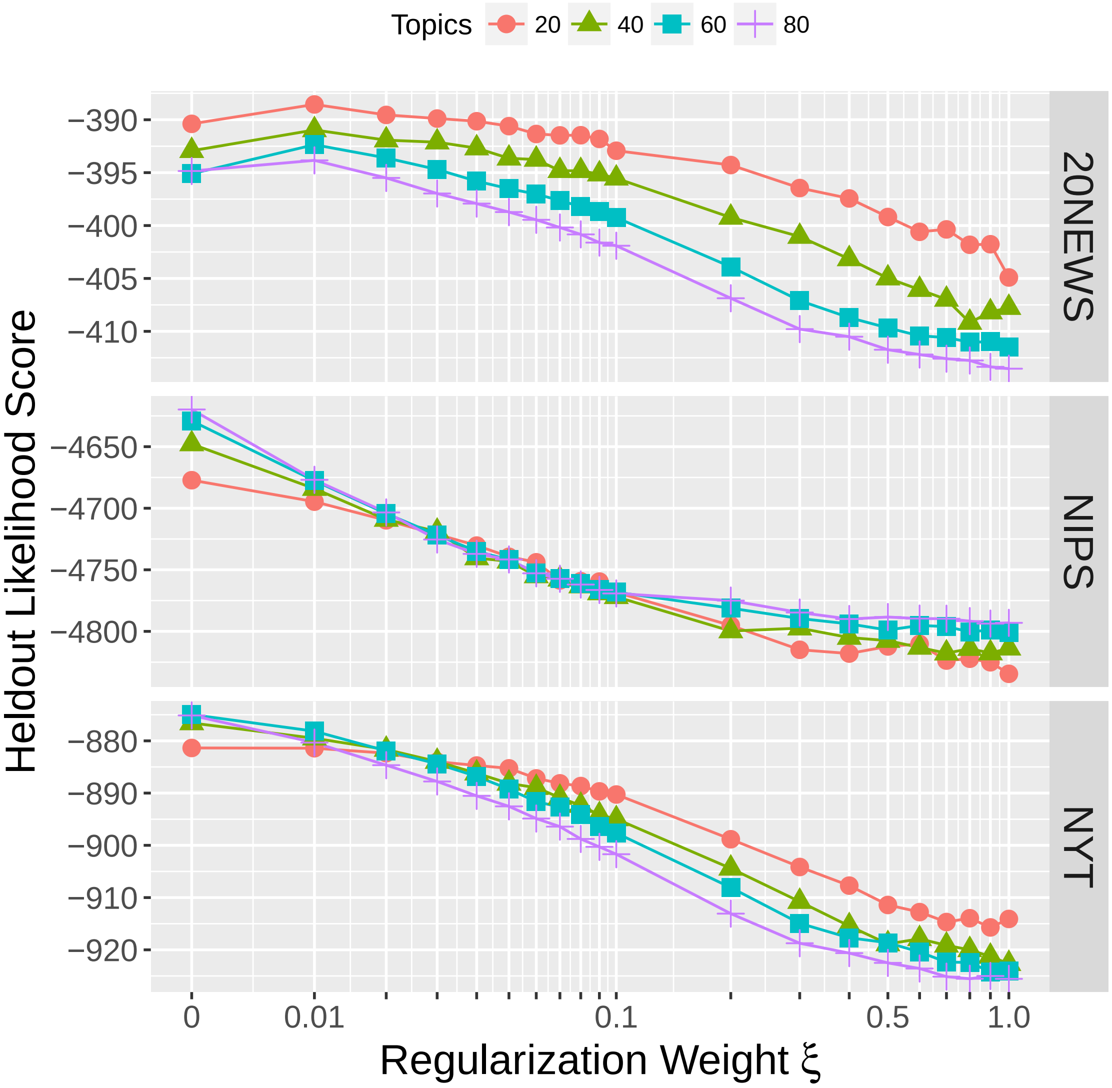


Figure B.1: Selection of ξ based on heldout likelihood score using ANCHOR-BETA on the development set. The value of $\xi = 0$ is equivalent to the unregularized anchor word algorithm.

B.2 Topic Interpretability Score using L_2 regularization

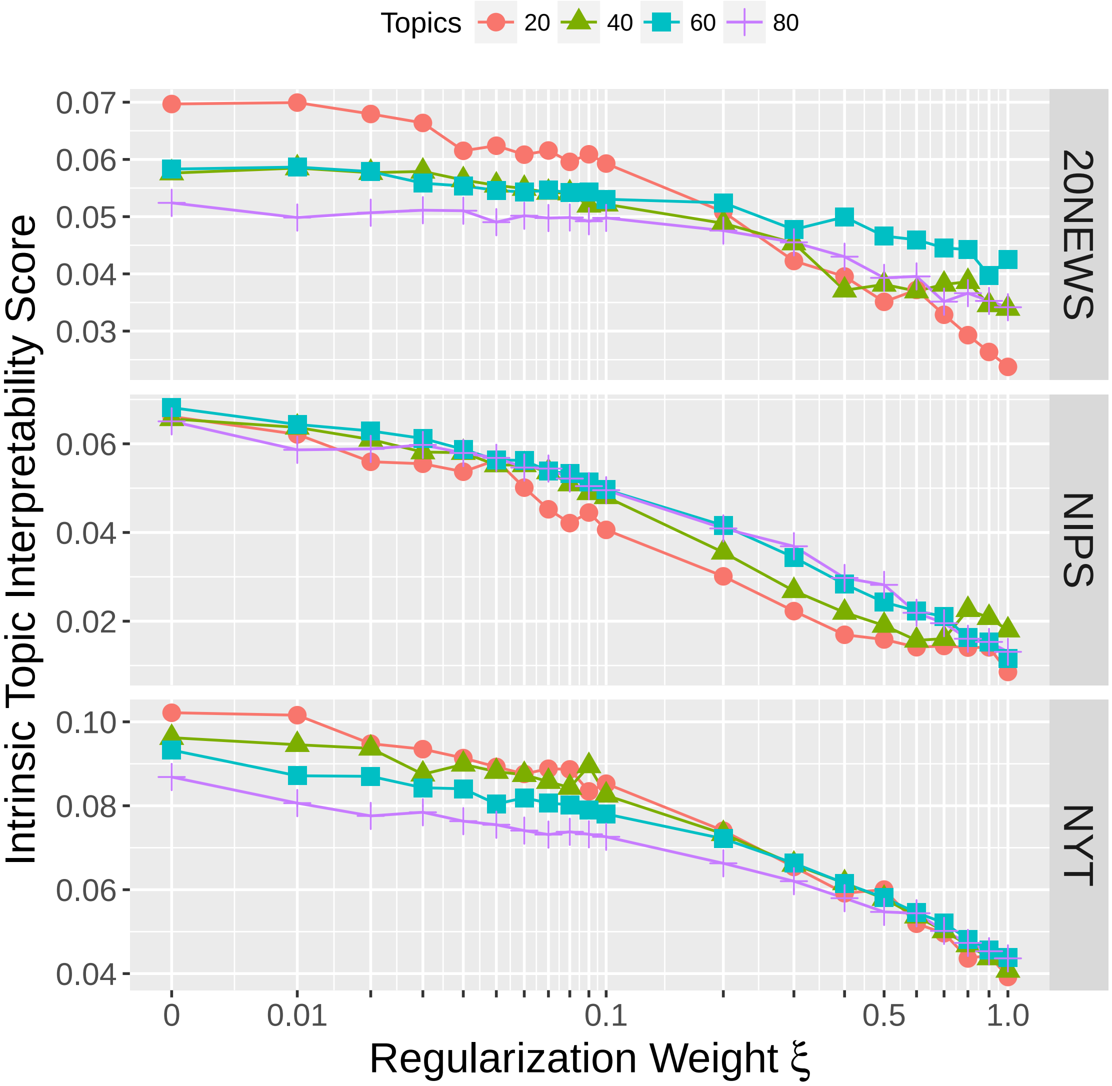


Figure B.2: Selection of ξ based on intrinsic topic interpretability score using ANCHOR-L_2 (bottom) on the development set. The value of $\xi = 0$ is equivalent to the unregularized anchor word algorithm.

Bibliography

Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *CoRR*, abs/1609.08675, 2016. URL http://arxiv.org/abs/1609.08675.

Ryan Adams, Zoubin Ghahramani, and Michael Jordan. Tree-structured stick breaking for hierarchical data. In *Proceedings of Advances in Neural Information Processing Systems*, pages 19–27, 2010.

Charu C. Aggarwal and Haixun Wang. *Text Mining in Social Networks*, pages 353–378. Springer US, Boston, MA, 2011. ISBN 978-1-4419-8462-3. doi: 10.1007/978-1-4419-8462-3_13. URL https://doi.org/10.1007/978-1-4419-8462-3_13.

Hirotogu Akaike. *Information Theory and an Extension of the Maximum Likelihood Principle*. Springer New York, 1998. URL https://doi.org/10.1007/978-1-4612-1694-0_15.

Nikolaos Aletras and Mark Stevenson. Evaluating topic coherence using distributional semantics, 2013.

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. Emotions from text: Machine learning for text-based emotion prediction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 579–586, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1220575.1220648. URL https://doi.org/10.3115/1220575.1220648.

Loulwah Alsumait, Daniel Barbará, James Gentle, and Carlotta Domeniconi. Topic significance ranking of lda generative models. In *In Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD*, pages 67–82, 2009.

Anima Anandkumar, Dean P. Foster, Daniel Hsu, Sham Kakade, and Yi-Kai Liu. A spectral algorithm for latent dirichlet allocation. In *NIPS*, pages 926–934, 2012a.

Animashree Anandkumar, Dean P. Foster, Daniel Hsu, Sham M. Kakade, and Yi-Kai Liu. Two svds suffice: Spectral decompositions for probabilistic topic modeling and latent dirichlet allocation. *CoRR*, abs/1204.6703, 2012b.

Animashree Anandkumar, Daniel Hsu, and Sham M. Kakade. A method of moments for mixture models and hidden Markov models. *Journal of Machine Learning Research - Proceedings Track*, 23:33.1–33.34, 2012c.

Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models, 2014.

Christophe Andrieu, Nando De Freitas, and et al. An introduction to mcmc for machine learning, 2003.

David Andrzejewski, Xiaojin Zhu, and Mark Craven. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 25–32, New York, NY, USA, 2009a. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553378. URL http://doi.acm.org/10.1145/1553374.1553378.

David Andrzejewski, Xiaojin Zhu, and Mark Craven. Incorporating domain knowledge into topic modeling via Dirichlet forest priors. In *Proceedings of the International Conference of Machine Learning*, 2009b.

David Andrzejewski, Xiaojin Zhu, Mark Craven, and Benjamin Recht. A framework for incorporating general domain knowledge into latent Dirichlet allocation using first-order logic. In *International Joint Conference on Artificial Intelligence*, 2011.

Sanjeev Arora, Rong Ge, Ravindran Kannan, and Ankur Moitra. Computing a nonnegative matrix factorization – provably. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 145–162, New York, NY, USA, 2012a. ACM. ISBN 978-1-4503-1245-5. doi: 10.1145/2213977.2213994. URL `http://doi.acm.org/10.1145/2213977.2213994`.

Sanjeev Arora, Rong Ge, and Ankur Moitra. Learning topic models — going beyond svd. *CoRR*, abs/1204.1956, 2012b.

Sanjeev Arora, Rong Ge, Yoni Halpern, David M. Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. A practical algorithm for topic modeling with provable guarantees. In *Proceedings of the International Conference of Machine Learning*, 2013.

Sanjeev Arora, Rong Ge, Frederic Koehler, Tengyu Ma, and Ankur Moitra. Provable algorithms for inference in topic models. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2859–2867, 2016. URL `http://jmlr.org/proceedings/papers/v48/arorab16.pdf`.

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *CoRR*, abs/1902.09229, 2019. URL `http://arxiv.org/abs/1902.09229`.

Arthur Asuncion, Padhraic Smyth, and Max Welling. Asynchronous distributed learning of topic models. In *Proceedings of Advances in Neural Information Processing Systems*, 2008.

Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *Found. Trends Mach. Learn.*, 4(1):1–106, January 2012. ISSN 1935-8237. doi: 10.1561/2200000015. URL `http://dx.doi.org/10.1561/2200000015`.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015. doi: 10.1371/journal.pone.0130140. URL `https://doi.org/10.1371/journal.pone.0130140`.

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015. URL `http://arxiv.org/abs/1511.00561`.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014. URL `http://arxiv.org/abs/1409.0473`. cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.

Jason Baldridge and Alexis Palmer. How well does active learning actually work?: Time-based evaluation of cost-reduction strategies for language documentation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 296–305, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-59-6. URL `http://dl.acm.org/citation.cfm?id=1699510.1699549`.

Georgios Balikas and Massih-Reza Amini. Twise at semeval-2016 task 4: Twitter sentiment classification. In *SemEval@NAACL-HLT*, 2016.

Richard G. Baraniuk, Volkan Cevher, Marco F. Duarte, and Chinmay Hegde. Model-based compressive sensing. *IEEE Trans. Inf. Theor.*, 56(4):1982–2001, April 2010. ISSN 0018-9448. doi: 10.1109/TIT.2010.2040894. URL `http://dx.doi.org/10.1109/TIT.2010.2040894`.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1023. URL `https://www.aclweb.org/anthology/P14-1023`.

Nicholas Beauchamp. Predicting and interpolating state-level polls using twitter textual data. *American Journal of Political Science*, 61(2):490–503, 2016. doi: 10.1111/ajps.12274. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/ajps.12274.

R. Bell, Y. Koren, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42: 30–37, 08 2009. ISSN 0018-9162. doi: 10.1109/MC.2009.263. URL doi.ieeecomputersociety.org/10.1109/MC.2009.263.

Richard.E Bellman. *Dynamic programming*. Princeton University Press, 1957.

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71, mar 1996. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=234285.234289.

Peter J. Bickel, Bo Li, Alexandre B. Tsybakov, and at el. Regularization in statistics. 2006.

José M. Bioucas-Dias, Antonio Plaza, Senior Member, Nicolas Dobigeon, Mario Parente, Qian Du, Senior Member, Paul Gader, and Jocelyn Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens*, pages 354–379, 2012.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *ArXiv e-prints*, 1 2016.

David M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, April 2012.

David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the International Conference of Machine Learning*, 2006.

David M. Blei and Jon D. McAuliffe. Supervised topic models. In *nips*. 2007.

David M. Blei, Andrew Ng, and Michael Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.

David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems*, page 2003. MIT Press, 2004.

David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *J. ACM*, 57(2):7:1–7:30, February 2010. ISSN 0004-5411. doi: 10.1145/1667053.1667056. URL http://doi.acm.org/10.1145/1667053.1667056.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-102-6. doi: 10.1145/1376616.1376746. URL http://doi.acm.org/10.1145/1376616.1376746.

Léon Bottou. Online algorithms and stochastic approximations. In *Online Learning and Neural Networks*. Cambridge University Press, Cambridge, UK, 1998.

Leon Bottou and Yann Le Cun. Large scale online learning. In *nips*. 2003.

Y. L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2559–2566, June 2010. doi: 10.1109/CVPR.2010.5539963.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. 2014.

Jordan Boyd-Graber and David M. Blei. PUTOP: Turning predominant senses into a topic model for WSD. In *Proceedings of the Workshop on Semantic Evaluation*, 2007.

Jordan Boyd-Graber and David M. Blei. Syntactic topic models. In *Proceedings of Advances in Neural Information Processing Systems*, 2008.

Jordan Boyd-Graber and David M. Blei. Multilingual topic models for unaligned text. In *Proceedings of Uncertainty in Artificial Intelligence*, 2009.

Jordan Boyd-Graber and Philip Resnik. Holistic sentiment analysis across languages: Multilingual supervised latent Dirichlet allocation. In *Proceedings of Empirical Methods in Natural Language Processing*, 2010.

Jordan Boyd-Graber, David M. Blei, and Xiaojin Zhu. A topic model for word sense disambiguation. In *Proceedings of Empirical Methods in Natural Language Processing*, 2007.

Michael Bryant and Erik Sudderth. Truly nonparametric online variational inference for hierarchical Dirichlet processes. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2708–2716. 2012. URL `http://books.nips.cc/papers/files/nips25/NIPS2012_1251.pdf`.

Wray Buntine. Estimating likelihoods for topic models. In *Asian Conference on Machine Learning*, page 0, Nanjing/China, November 2009.

Dallas Card, Chenhao Tan, and Noah A. Smith. Neural models for documents with metadata. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2031–2040. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/P18-1189`.

Patricia A. Cavazos-Rehg, Melissa J. Krauss, Shaina Sowles, Sarah Connolly, Carlos Rosas, Meghana Bharadwaj, and Laura J. Bierut. A content analysis of depression-related tweets. *Comput. Hum. Behav.*, 54(C):351–357, January 2016. ISSN 0747-5632. doi: 10.1016/j.chb.2015.08.023. URL `http://dx.doi.org/10.1016/j.chb.2015.08.023`.

Miriam Cha, Youngjune Gwon, and H. T. Kung. Twitter geolocation and regional classification via sparse coding. In *AAAI Publications, Ninth International AAAI Conference on Web and Social Media*, 2015.

Tsung-han Chan, Wing-kin Ma, Chong-yung Chi, Senior Member, and Yue Wang. A convex analysis framework for blind separation of non-negative sources. *IEEE Trans. Signal Process*, pages 5120–5134, 2008.

Allison Chaney and David Blei. Visualizing topic models. In *International AAAI Conference on Weblogs and Social Media*, 2012. URL `https://www.aaai.org/ocs/index.php/ICWSM/ICWSM12/paper/view/4645/5021`.

H. Chang, J. Han, C. Zhong, A. M. Snijders, and J. Mao. Unsupervised transfer learning via multi-scale convolutional sparse coding for biomedical applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1182–1194, May 2018. ISSN 0162-8828. doi: 10.1109/TPAMI.2017.2656884.

Jonathan Chang and David M. Blei. Hierarchical relational models for document networks. *The Annals of Applied Statistics*, 2010.

Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *Proceedings of Advances in Neural Information Processing Systems*, 2009. URL `http://www.cs.colorado.edu/~jbg/docs/nips2009-rtl.pdf`.

O. Chapelle, B. Scholkopf, and A. Zien, Eds. Semi-supervised learning (chapelle, o. et al., eds.; 2006) [book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, March 2009. ISSN 1045-9227. doi: 10.1109/TNN.2009.2015974.

Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates, and José Luis Marroquín. Searching in metric spaces. *ACM Comput. Surv.*, 33(3):273–321, September 2001. ISSN 0360-0300. doi: 10.1145/502807.502808. URL `http://doi.acm.org/10.1145/502807.502808`.

Bernard Chazelle. Computational geometry: A retrospective. In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, STOC '94, pages 75–94, New York, NY, USA, 1994. ACM. ISBN 0-89791-663-8. doi: 10.1145/195058.195110. URL `http://doi.acm.org/10.1145/195058.195110`.

Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 2722–2730, Washington, DC, USA, 2015a. IEEE Computer Society. ISBN 978-1-4673-8391-2. doi: 10.1109/ICCV.2015.312. URL `http://dx.doi.org/10.1109/ICCV.2015.312`.

L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, April 2018. ISSN 0162-8828. doi: 10.1109/TPAMI.2017.2699184.

Tiffany Yu-Han Chen, Lenin Ravindranath, Shuo Deng, Paramvir Bahl, and Hari Balakrishnan. Glimpse: Continuous, real-time object recognition on mobile devices. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, SenSys '15, pages 155–168, New York, NY, USA, 2015b. ACM. ISBN 978-1-4503-3631-4. doi: 10.1145/2809695.2809711. URL `http://doi.acm.org/10.1145/2809695.2809711`.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 2180–2188, USA, 2016. Curran Associates Inc. ISBN 978-1-5108-3881-9. URL `http://dl.acm.org/citation.cfm?id=3157096.3157340`.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL `https://www.aclweb.org/anthology/D14-1179`.

Gregory C. Chow. Maximum-likelihood estimation of misspecified models. *Economic Modelling*, 1 (2):134 – 138, 1984. URL `http://www.sciencedirect.com/science/article/pii/0264999384900014`.

Jason Chuang, Christopher D. Manning, and Jeffrey Heer. Termite: Visualization techniques for assessing textual topic models. In *Advanced Visual Interfaces*, 2012. URL `http://vis.stanford.edu/papers/termite`.

Shay B. Cohen and Michael Collins. A provably correct learning algorithm for latent-variable PCFGs. In *Proceedings of the Association for Computational Linguistics*, 2014.

Shay B. Cohen and Noah A. Smith. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2009.

Shay B. Cohen, Karl Stratos, Michael Collins, Dean P. Foster, and Lyle Ungar. Experiments with spectral learning of latent-variable PCFGs. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2013.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390177. URL `http://doi.acm.org/10.1145/1390156.1390177`.

Arthur Cook. *Forecasting for the Pharmaceutical Industry*. London: Gower, 2015.

Kenneth Cukier. Big data: A revolution that will transform how we live, work, and think. *Viktor Mayer-Schönberger*, 2014.

Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 208–215, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390183. URL `http://doi.acm.org/10.1145/1390156.1390183`.

Scott Deerwester, Susan Dumais, Thomas Landauer, George Furnas, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.

J. Deng, W. Dong, R. Socher, L. J. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848.

Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, Jan 2001. ISSN 1573-0565. doi: 10.1023/A:1007612920971. URL `https://doi.org/10.1023/A:1007612920971`.

C. H. Q. Ding, T. Li, and M. I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):45–55, Jan 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2008.277.

Chris Ding, Xiaofeng He, and Horst D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *in SIAM International Conference on Data Mining*, 2005.

Chris Ding, Tao Li, and Wei Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Comput. Stat. Data Anal.*, 52(8):3913–3927, April 2008. ISSN 0167-9473. doi: 10.1016/j.csda.2008.01.011. URL `http://dx.doi.org/10.1016/j.csda.2008.01.011`.

W. Ding, P. Ishwar, and V. Saligrama. Most large topic models are approximately separable. In *2015 Information Theory and Applications Workshop (ITA)*, pages 199–203, Feb 2015. doi: 10.1109/ITA.2015.7308989.

Weicong Ding, Mohammad Hossein Rohban, Prakash Ishwar, and Venkatesh Saligrama. Topic discovery through data dependent and random projections. In *Proceedings of the International Conference of Machine Learning*, 2013a.

Weicong Ding, Mohammad Hossein Rohban, Prakash Ishwar, and Venkatesh Saligrama. A new geometric approach to latent topic modeling and discovery. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013b.

Pinar Donmez and Jaime G. Carbonell. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pages 619–628, 2008.

David Donoho and Victoria Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 1141–1148. MIT Press, 2004. URL `http://papers.nips.cc/paper/2463-when-does-non-negative-matrix-factorization-give-a-correct-decomposition-int pdf`.

B. Du, Z. Wang, L. Zhang, L. Zhang, W. Liu, J. Shen, and D. Tao. Exploring representativeness and informativeness for active learning. *IEEE Transactions on Cybernetics*, 47(1):14–26, Jan 2017. ISSN 2168-2267. doi: 10.1109/TCYB.2015.2496974.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. Technical Report UCB/EECS-2010-24, EECS Department, University of California, Berkeley, Mar 2010. URL `http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-24.html`.

P. Duygulu, K. Barnard, J. F. G. de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Computer Vision — ECCV 2002*, pages 97–112, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-47979-6.

Jacob Eisenstein, Amr Ahmed, and Eric P. Xing. Sparse additive generative models of text. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1041–1048, New York, NY, USA, 2011. ACM. URL `http://www.icml-2011.org/papers/534_icmlpaper.pdf`.

Jacob Eisenstein, Duen Horng Chau, Aniket Kittur, and Eric Xing. TopicViz: interactive topic exploration in document collections. In *International Conference on Human Factors in Computing Systems*, 2012.

T. Eltoft, Taesu Kim, and Te-Won Lee. On the multivariate laplace distribution. 2006.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res.*, 11:625–660, March 2010. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1756006.1756025`.

E. Erosheva, S. Fienberg, and J. Lafferty. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Science*, 97(22):11885–11892, 2004.

Jerry Alan Fails and Dan R. Olsen, Jr. Interactive machine learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, pages 39–45, New York, NY, USA, 2003. ACM. ISBN 1-58113-586-6. doi: 10.1145/604045.604056. URL `http://doi.acm.org/10.1145/604045.604056`.

Thiago Faleiros and Alneu Lopes. On the equivalence between algorithms for non-negative matrix factorization and latent dirichlet allocation. In *ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2016.

Christiane Fellbaum. Wordnet and wordnets. In Alex Barber, editor, *Encyclopedia of Language and Linguistics*, pages 2–665. Elsevier, 2005.

Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural Comput.*, 21(3):793–830, March 2009. ISSN 0899-7667. doi: 10.1162/neco.2008.04-08-771. URL http://dx.doi.org/10.1162/neco.2008.04-08-771.

D. P. Foster, J. Rodu, and L. H. Ungar. Spectral dimensionality reduction for hmms. 2012.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28:2000, 1998.

Johannes Furnkranz. A study using n-gram features for text categorization, 1998.

Mark Galassi, Jim Davies, James Theiler, Brian Gough, Gerard Jungman, Michael Booth, and Fabrice Rossi. *Gnu Scientific Library: Reference Manual*. Network Theory Ltd., 2003. ISBN 0954161734.

N. P. Galatsanos and A. K. Katsaggelos. Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation. *IEEE Transactions on Image Processing*, 1(3): 322–336, Jul 1992. ISSN 1057-7149. doi: 10.1109/83.148606.

Matthew Gardner, Joshua Lutes, Jeff Lund, Josh Hansen, Dan Walker, Eric Ringger, and Kevin Seppi. The topic browser: An interactive tool for browsing topic models. In *Proceedings of Advances in Neural Information Processing Systems*, 2010.

Caroline Gasperin. Active learning for anaphora resolution. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, HLT '09, pages 1–8, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1564131.1564133.

Rainer Gemulla, Erik Nijkamp, Peter J. Haas, and Yannis Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 69–77, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0813-7. doi: 10.1145/2020408.2020426. URL http://doi.acm.org/10.1145/2020408.2020426.

Nicolas Gillis. Sparse and unique nonnegative matrix factorization through data preprocessing. *J. Mach. Learn. Res.*, 13(1):3349–3386, November 2012. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=2503308.2503349.

Nicolas Gillis. The why and how of nonnegative matrix factorization. In *Regularization, Optimization, Kernels, and Support Vector Machines. Chapman & Hall/CRC*, 2014.

Federico Girosi, Michael Jones, and Tomaso Poggio. Regularization theory and neural networks architectures. *Neural Comput.*, 7(2):219–269, March 1995. ISSN 0899-7667. doi: 10.1162/neco.1995.7.2.219. URL http://dx.doi.org/10.1162/neco.1995.7.2.219.

Gene H. Goluba, Michael Heathb, and Grace Wahbac. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21, 1979.

A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, May 2013. doi: 10.1109/ICASSP.2013.6638947.

Brynjar Gretarsson, John O'Donovan, Svetlin Bostandjiev, Tobias Höllerer, Arthur Asuncion, David Newman, and Padhraic Smyth. Topicnets: Visual analysis of large text corpora with topic modeling. *ACM Trans. Intell. Syst. Technol.*, 3(2):23:1–23:26, February 2012. ISSN 2157-6904. doi: 10.1145/2089094.2089099. URL `http://doi.acm.org/10.1145/2089094.2089099`.

Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl 1):5228–5235, 2004.

Thomas L. Griffiths, Mark Steyvers, David M. Blei, and Joshua B. Tenenbaum. Integrating topics and syntax. In *Proceedings of Advances in Neural Information Processing Systems*. 2005.

Justin Grimmer. We are all social scientists now: How big data, machine learning, and causal inference work together. *PS: Political Science and Politics*, 48(1):80–83, 2015. doi: 10.1017/S1049096514001784.

Yuhong Guo and Russ Greiner. Optimistic active learning using mutual information. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 823–829, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc. URL `http://dl.acm.org/citation.cfm?id=1625275.1625408`.

Yuhong Guo and Dale Schuurmans. Discriminative batch mode active learning. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pages 593–600, USA, 2007. Curran Associates Inc. ISBN 978-1-60560-352-0. URL `http://dl.acm.org/citation.cfm?id=2981562.2981637`.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=944919.944968`.

Robbie Haertel, Eric Ringger, Kevin Seppi, James Carroll, and Peter McClanahan. Assessing the costs of sampling methods in active learning for annotation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 65–68. Association for Computational Linguistics, 2008a.

Robbie Haertel, Eric K. Ringger, Kevin D. Seppi, James L. Carroll, and Peter McClanahan. Assessing the costs of sampling methods in active learning for annotation. In *Proceedings of the Association for Computational Linguistics*, 2008b.

Najeh Hajlaoui, David Kolovratnik, Jaakko Väyrynen, Ralf Steinberger, and Daniel Varga. Dcep -digital corpus of the european parliament. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may 2014. European Language Resources Association (ELRA). ISBN 978-2-9517408-8-4.

Zellig S. Harris. Distributional structure. *WORD*, 10(2-3):146–162, 1945. URL `https://doi.org/10.1080/00437956.1954.11659520`.

Richard A. Harshman and Margaret E. Lundy. Parafac: Parallel factor analysis. *Comput. Stat. Data Anal.*, 18(1):39–72, August 1994. ISSN 0167-9473. doi: 10.1016/0167-9473(94)90132-5. URL `http://dx.doi.org/10.1016/0167-9473(94)90132-5`.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2009.

Marti A. Hearst. Untangling text data mining. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pages 3–10, Stroudsburg, PA, USA, 1999. Association for Computational Linguistics. ISBN 1-55860-609-3. doi: 10.3115/1034678.1034679. URL `https://doi.org/10.3115/1034678.1034679`.

G. Heinrich. Parameter estimation for text analysis. Technical report, 2004. http://www.arbylon.net/publications/text-est.pdf.

G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, Nov 2012. ISSN 1053-5888. doi: 10.1109/MSP.2012.2205597.

Geoffrey E. Hinton and Ruslan R Salakhutdinov. Replicated softmax: an undirected topic model. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1607–1614. Curran Associates, Inc., 2009. URL `http://papers.nips.cc/paper/3856-replicated-softmax-an-undirected-topic-model.pdf`.

Matthew Hoffman, David M. Blei, and Francis Bach. Online learning for latent Dirichlet allocation. In *Proceedings of Advances in Neural Information Processing Systems*, 2010.

Matthew Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. In *Journal of Machine Learning Research*, 2013.

Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence*, 1999a.

Thomas Hofmann. Probabilistic latent semantic indexing. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999b.

Steven C. H. Hoi, Rong Jin, and Michael R. Lyu. Large-scale text categorization by batch mode active learning. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 633–642, New York, NY, USA, 2006. ACM. ISBN 1-59593-323-9. doi: 10.1145/1135777.1135870. URL `http://doi.acm.org/10.1145/1135777.1135870`.

Enamul Hoque and Giuseppe Carenini. Convisit: Interactive topic modeling for exploring asynchronous online conversations. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, 2015. ISBN 978-1-4503-3306-1.

Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.*, 5: 1457–1469, December 2004. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1005332.1044709`.

Xia Hu, Nan Sun, Chao Zhang, and Tat-Seng Chua. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 919–928, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-512-3. doi: 10.1145/1645953.1646071. URL `http://doi.acm.org/10.1145/1645953.1646071`.

Yuening Hu and Jordan Boyd-Graber. Efficient tree-based topic modeling. In *Proceedings of the Association for Computational Linguistics*, 2012.

Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. Interactive topic modeling. *Mach. Learn.*, 95(3):423–469, June 2014a. ISSN 0885-6125. URL `http://dx.doi.org/10.1007/s10994-013-5413-0`.

Yuening Hu, Ke Zhai, Vlad Eidelman, and Jordan Boyd-Graber. Polylingual tree-based topic models for translation domain adaptation. In *Proceedings of the Association for Computational Linguistics*, 2014b.

Furong Huang, Nikos Karampatziakis, Paul Mineiro, Sergiy Matusevych, and Animashree Anandkumar. Distributed latent dirichlet allocation via tensor factorization. In *NIPS Optimization for Machine Learning Workshop*, 12 2014a.

Furong Huang, U. N. Niranjan, Mohammad Umar Hakeem, Animashree An, and kumar. Online tensor methods for learning latent variable models. *Journal of Machine Learning Research*, 16(86):2797–2835, 2015. URL `http://jmlr.org/papers/v16/huang15a.html`.

Junzhou Huang, Tong Zhang, and Dimitris Metaxas. Learning with structured sparsity. *J. Mach. Learn. Res.*, 12:3371–3412, November 2011. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1953048.2078213`.

S. Huang, R. Jin, and Z. Zhou. Active learning by querying informative and representative examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(10):1936–1949, Oct 2014b. ISSN 0162-8828. doi: 10.1109/TPAMI.2014.2307881.

Zhengxing Huang, Wei Dong, Lei Ji, Chenxi Gan, Xudong Lu, and Huilong Duan. Discovery of clinical pathway patterns from event logs using probabilistic topic models. *J. of Biomedical Informatics*, 47: 39–57, February 2014c. ISSN 1532-0464. doi: 10.1016/j.jbi.2013.09.003. URL `http://dx.doi.org/10.1016/j.jbi.2013.09.003`.

Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. A neural network for factoid question answering over paragraphs. In *Proceedings of Empirical Methods in Natural Language Processing*, 2014.

Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 433–440, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553431. URL `http://doi.acm.org/10.1145/1553374.1553431`.

Jagadeesh Jagarlamudi and Hal Daumé III. Extracting multilingual topics from unaligned corpora. In *ecir*, Milton Keynes, United Kingdom, 2010.

Jagadeesh Jagarlamudi, Hal Daumé III, and Raghavendra Udupa. Incorporating lexical priors into topic models. In *EACL*, 2012.

Neal Jean, Marshall Burke, Michael Xie, W. Matthew Davis, David B. Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794, 2016. ISSN 0036-8075. doi: 10.1126/science.aaf7894. URL `https://science.sciencemag.org/content/353/6301/790`.

Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, and Francis Bach. Proximal methods for hierarchical sparse coding. *J. Mach. Learn. Res.*, 12:2297–2334, July 2011. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1953048.2021074`.

Friedman Jerome, Hastie Trevor, and Tibshirani Rob. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 2010.

S. Jiang, X. Qian, J. Shen, Y. Fu, and T. Mei. Author topic model-based collaborative filtering for personalized poi recommendations. *IEEE Transactions on Multimedia*, 17(6):907–918, June 2015. ISSN 1520-9210. doi: 10.1109/TMM.2015.2417506.

Nitin Jindal and Bing Liu. Opinion spam and analysis. In *Proceedings of First ACM International Conference on Web Search and Data Mining*, 2008.

Yohan Jo and Alice H. Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of ACM International Conference on Web Search and Data Mining*, 2011.

Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Machine Learning: ECML-98*, pages 137–142, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. ISBN 978-3-540-69781-7.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2007.

I.T. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001. URL `http://www.scipy.org/`.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999. ISSN 0885-6125. doi: 10.1023/A:1007665907178.

Ata Kabán. On bayesian classification with laplace priors. *Pattern Recogn. Lett.*, 28(10):1271–1282, July 2007. ISSN 0167-8655. doi: 10.1016/j.patrec.2007.02.010. URL `http://dx.doi.org/10.1016/j.patrec.2007.02.010`.

Devarajan Karthik. Nonnegative matrix factorization: An analytical and interpretive tool in computational biology. In *PLoS Computational Biology*, 2008.

Hyunsoo Kim and Haesun Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, June 2007. ISSN 1367-4803. doi: 10.1093/bioinformatics/btm134. URL `http://dx.doi.org/10.1093/bioinformatics/btm134`.

Diederik P. Kingma, Danilo J. Rezende, Shakir Mohamed, and Max Welling. Semi-supervised learning with deep generative models. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3581–3589, Cambridge, MA, USA, 2014. MIT Press. URL `http://dl.acm.org/citation.cfm?id=2969033.2969226`.

Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Inf. Comput.*, 132(1):1–63, January 1997. ISSN 0890-5401. doi: 10.1006/inco.1996.2612. URL `http://dx.doi.org/10.1006/inco.1996.2612`.

A. Kumar, V. Sindhwani, and P. Kambadur. Fast conical hull algorithms for near-separable non-negative matrix factorization. *jmlr*, 2013.

Alexandros Labrinidis and H. V. Jagadish. Challenges and opportunities with big data. *Proc. VLDB Endow.*, 5(12):2032–2033, August 2012. ISSN 2150-8097. doi: 10.14778/2367502.2367572. URL `http://dx.doi.org/10.14778/2367502.2367572`.

Simon Lacoste-Julien, Fei Sha, and Michael I. Jordan. Disclda: Discriminative learning for dimensionality reduction and classification. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 897–904. Curran Associates, Inc., 2009. URL `http://papers.nips.cc/paper/3599-disclda-discriminative-learning-for-dimensionality-reduction-and-classificat pdf`.

Ken Lang. 20 newsgroups data set, 2007. http://www.ai.mit.edu/people/jrennie/20Newsgroups/.

John Langford, Lihong Li, and Tong Zhang. Sparse online learning via truncated gradient. *J. Mach. Learn. Res.*, 10:777–801, June 2009. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1577069.1577097.

Hugo Larochelle and Stanislas Lauly. A neural autoregressive topic model. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2708–2716. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4613-a-neural-autoregressive-topic-model.pdf.

Rémi Lebret and Ronan Collobert. Word embeddings through hellinger pca. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–490. Association for Computational Linguistics, 2014. doi: 10.3115/v1/E14-1051. URL http://aclweb.org/anthology/E14-1051.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. *Deep Learning*, volume 521. 2015. URL https://doi.org/10.1038/nature14539.

Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. 1999.

Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press, 2001. URL http://papers.nips.cc/paper/1861-algorithms-for-non-negative-matrix-factorization.pdf.

Moontae Lee and David Mimno. Low-dimensional embeddings for interpretable anchor-based topic inference. In *Proceedings of Empirical Methods in Natural Language Processing*, 2014a.

Moontae Lee and David Mimno. Low-dimensional embeddings for interpretable anchor-based topic inference. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1319–1328, Doha, Qatar, October 2014b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D14-1138.

Moontae Lee, David Bindel, and David Mimno. Robust spectral inference for joint stochastic matrix factorization. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 2710–2718, Cambridge, MA, USA, 2015. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969442.2969542.

Moontae Lee, David Bindel, and David Mimno. From correlation to hierarchy: Practical topic modeling via spectral inference. In *Proceedings of the 12th INFORMS Workshop on Data Mining and Decision Analytics*, Houston, USA, 10 2017a.

Moontae Lee, David Bindel, and David M. Mimno. Prior-aware dual decomposition: Document-specific topic inference for spectral topic models. *CoRR*, abs/1711.07065, 2017b. URL http://arxiv.org/abs/1711.07065.

Tak Yeon Lee, Alison Smith, Kevin Seppi, Niklas Elmqvist, Jordan Boyd-Graber, and Leah Findlater. The human touch: How non-expert users perceive, interpret, and fix topic models. *International Journal of Human-Computer Studies*, 105:28 – 42, 2017c. ISSN 1071-5819. doi: https://doi.org/10.1016/j.ijhcs.2017.03.007. URL http://www.sciencedirect.com/science/article/pii/S1071581917300472.

Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

Omer Levy and Yoav Goldberg. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 171–180. Association for Computational Linguistics, 2014a. doi: 10.3115/v1/W14-1618. URL http://aclweb.org/anthology/W14-1618.

Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2177–2185, Cambridge, MA, USA, 2014b. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969033.2969070.

David D. Lewis. *Reuters-21578 text categorization test collection, distribution 1.0*, 1997. URL http://www.daviddlewis.com/resources/testcollections/reuters21578.

David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994a.

David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 3–12, New York, NY, USA, 1994b. Springer-Verlag New York, Inc. ISBN 0-387-19889-X. URL http://dl.acm.org/citation.cfm?id=188490.188495.

Aaron Q Li, Amr Ahmed, Sujith Ravi, and Alexander J Smola. Reducing the sampling complexity of topic models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 891–900. ACM, 2014.

Fei-Fei Li and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 524–531, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2. doi: 10.1109/CVPR.2005.16. URL http://dx.doi.org/10.1109/CVPR.2005.16.

Xiao Li, Da Kuang, and Charles X. Ling. Active learning for hierarchical text classification. In *Proceedings of the 16th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining - Volume Part I*, PAKDD'12, pages 14–25, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-30216-9. doi: 10.1007/978-3-642-30217-6_2. URL http://dx.doi.org/10.1007/978-3-642-30217-6_2.

Yitan Li, Linli Xu, Fei Tian, Liang Jiang, Xiaowei Zhong, and Enhong Chen. Word embedding revisited: A new representation learning and explicit matrix factorization perspective. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 3650–3656. AAAI Press, 2015. ISBN 978-1-57735-738-4. URL http://dl.acm.org/citation.cfm?id=2832747.2832758.

Maxwell W. Libbrecht and William Stafford Noble. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 2015.

Chinghway Lim and Bin Yu. Estimation stability with cross-validation (escv). *Journal of Computational and Graphical Statistics*, 25, 2016.

Robert V. Lindsey, William P. Headden, III, and Michael J. Stipicevic. A phrase-discovering topic model using hierarchical pitman-yor processes. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 214–222, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2390948.2390975.

Bing Liu. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012. ISBN 1608458849, 9781608458844.

Lin Liu, Lin Tang, Wen Dong, Shaowen Yao, and Wei Zhou. An overview of topic modeling and its current applications in bioinformatics. volume 5. Springer International Publishing, 09 2016. URL `https://www.ncbi.nlm.nih.gov/pubmed/27652181`.

Shixia Liu, Michelle X. Zhou, Shimei Pan, Weihong Qian, Weijia Cai, and Xiaoxiao Lian. Interactive, topic-based visual text summarization and analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 543–552, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-512-3. doi: 10.1145/1645953.1646023. URL `http://doi.acm.org/10.1145/1645953.1646023`.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 9 1999. doi: 10.1109/ICCV.1999.790410.

Yue Lu, Qiaozhu Mei, and Chengxiang Zhai. Investigating task performance of probabilistic topic models: An empirical study of plsa and lda. *Inf. Retr.*, 14(2):178–203, April 2011. ISSN 1386-4564. doi: 10.1007/s10791-010-9141-9. URL `http://dx.doi.org/10.1007/s10791-010-9141-9`.

Jeffrey Lund, Connor Cook, Kevin Seppi, and Jordan Boyd-Graber. Tandem anchoring: a multiword anchor approach for interactive topic modeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 896–905, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1083. URL `https://www.aclweb.org/anthology/P17-1083`.

Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208, 1996. URL `https://doi.org/10.3758/BF03204766`.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf`.

Wenhan Luo, Björn Stenger, Xiaowei Zhao, and Tae-Kyun Kim. Automatic topic discovery for multi-object tracking. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 3820–3826. AAAI Press, 2015. ISBN 0-262-51129-0. URL `http://dl.acm.org/citation.cfm?id=2888116.2888246`.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1166. URL `https://www.aclweb.org/anthology/D15-1166`.

J. Macqueen. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2272–2279, Sept 2009. doi: 10.1109/ICCV.2009.5459452.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, pages 19–60, 2010.

Julien Mairal, Francis Bach, and Jean Ponce. Sparse modeling for image and vision processing. *Found. Trends. Comput. Graph. Vis.*, 8(2-3):85–283, December 2014. ISSN 1572-2740. doi: 10.1561/0600000058. URL `http://dx.doi.org/10.1561/0600000058`.

A. Majumdar and R. K. Ward. Non-convex group sparsity: Application to color imaging. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 469–472, March 2010. doi: 10.1109/ICASSP.2010.5495703.

Polina Mamoshina, Armando Vieira, Evgeny Putin, and Alex Zhavoronkov. Applications of deep learning in biomedicine. *Molecular Pharmaceutics*, 13(5):1445–1454, 2016. doi: 10.1021/acs.molpharmaceut.5b00982. URL `https://doi.org/10.1021/acs.molpharmaceut.5b00982`. PMID: 27007977.

James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Byers. Big data: The next frontier for innovation, competition, and productivity. *McKinsey Global Institute*, 2011.

r. b. Marimont and m. b. shapiro. Nearest neighbour searches and the curse of dimensionality. *IMA Journal of Applied Mathematics*, 24(1):59, 1979. doi: 10.1093/imamat/24.1.59. URL `+http://dx.doi.org/10.1093/imamat/24.1.59`.

Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 188–191, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119176.1119206. URL `https://doi.org/10.3115/1119176.1119206`.

Andrew McCallum and Kamal Nigam. Employing em and pool-based active learning for text classification. In *Proceedings of the International Conference of Machine Learning*, pages 350–358, 1998.

Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://www.cs.umass.edu/ mccallum/mallet, 2002.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA, 2013a. Curran Associates Inc. URL `http://dl.acm.org/citation.cfm?id=2999792.2999959`.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013b.

David Mimno and Andrew McCallum. Mining a digital library for influential authors. In *JCDL '07*, New York, NY, USA, 2007. ACM.

David Mimno and Andrew McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, UAI'08, pages 411–418, Arlington, Virginia, United States, 2008. AUAI Press. ISBN 0-9749039-4-9. URL `http://dl.acm.org/citation.cfm?id=3023476.3023525`.

David Mimno, Wei Li, and Andrew McCallum. Mixtures of hierarchical topics with pachinko allocation. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 633–640, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. doi: 10.1145/1273496.1273576. URL `http://doi.acm.org/10.1145/1273496.1273576`.

David Mimno, Hanna Wallach, and Andrew McCallum. Gibbs sampling for logistic normal topic models with graph-based priors. In *NIPS 2008 Workshop on Analyzing Graphs: Theory and Applications*, 2008.

David Mimno, Hanna Wallach, Jason Naradowsky, David Smith, and Andrew McCallum. Polylingual topic models. In *Proceedings of Empirical Methods in Natural Language Processing*, 2009.

David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 262–272, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4. URL `http://dl.acm.org/citation.cfm?id=2145432.2145462`.

Thomas P. Minka. Bayesian inference, entropy, and the multinomial distribution, 2000a.

Thomas P. Minka. Estimating a dirichlet distribution. Technical report, Microsoft, 2000b. http://research.microsoft.com/en-us/um/people/minka/papers/dirichlet/.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiao-Dan Zhu, and Colin Cherry. Semeval-2016 task 6: Detecting stance in tweets. In *SemEval@NAACL-HLT*, 2016.

Ankur Moitra. An almost optimal algorithm for computing nonnegative rank. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 1454–1464, Philadelphia, PA, USA, 2013. Society for Industrial and Applied Mathematics. ISBN 978-1-611972-51-1. URL `http://dl.acm.org/citation.cfm?id=2627817.2627921`.

Jean Morales, Charles A. Micchelli, and Massimiliano Pontil. A family of penalty functions for structured sparsity. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1612–1623. Curran Associates, Inc., 2010. URL `http://papers.nips.cc/paper/4137-a-family-of-penalty-functions-for-structured-sparsity.pdf`.

Robert Moskovitch, Nir Nissim, Dima Stopel, Clint Feher, Roman Englert, and Yuval Elovici. Improving the detection of unknown computer worms activity using active learning. In Joachim Hertzberg, Michael Beetz, and Roman Englert, editors, *KI 2007: Advances in Artificial Intelligence*, pages 489–493, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-74565-5.

Kevin P Murphy. *Machine learning: a probabilistic perspective*, page 744. MIT Press, 2012.

Chris Musialek, Philip Resnik, and S. Andrew Stavisky. Using text analytic techniques to create efficiencies in analyzing qualitative data: A comparison between traditional content analysis and a topic modeling approach. In *In American Association for Public Opinion Research*, 2016.

Ramesh Nallapati, William Cohen, and John Lafferty. Parallelized variational EM for latent Dirichlet allocation: An experimental evaluation of speed and scalability. In *ICDMW*, 2007.

Ramesh M. Nallapati, Amr Ahmed, Eric P. Xing, and William W. Cohen. Joint latent topic models for text and citations. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 542–550, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. doi: 10.1145/1401890.1401957. URL `http://doi.acm.org/10.1145/1401890.1401957`.

Radford M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993.

Radford M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

David Newman, Chaitanya Chemudugunta, and Padhraic Smyth. Statistical entity-topic models. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 680–686, New York, NY, USA, 2006. ACM. ISBN 1-59593-339-5. doi: http://doi.acm.org/10.1145/1150402.1150487.

David Newman, Sarvnaz Karimi, and Lawrence Cavedon. External evaluation of topic models. In *Proceedings of the Aurstralasian Document Computing Symposium*, 2009.

David Newman, Timothy Baldwin, Lawrence Cavedon, Eric Huang, Sarvnaz Karimi, David Martinez, Falk Scholer, and Justin Zobel. Invited paper: Visualizing search results and document collections using topic maps. *Web Semant.*, 8:169–175, July 2010a. ISSN 1570-8268. doi: http://dx.doi.org/10.1016/j.websem.2010.03.005. URL `http://dx.doi.org/10.1016/j.websem.2010.03.005`.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2010b.

David Newman, Edwin V Bonilla, and Wray Buntine. Improving topic coherence with regularized topic models. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 496–504. Curran Associates, Inc., 2011. URL `http://papers.nips.cc/paper/4291-improving-topic-coherence-with-regularized-topic-models.pdf`.

Andrew Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the International Conference of Machine Learning*, 2004.

Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pages 841–848, Cambridge, MA, USA, 2001. MIT Press. URL `http://dl.acm.org/citation.cfm?id=2980539.2980648`.

Cam-Tu Nguyen, De-Chuan Zhan, and Zhi-Hua Zhou. Multi-modal image annotation with multi-instance multi-label lda. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 1558–1564. AAAI Press, 2013a. ISBN 978-1-57735-633-2. URL `http://dl.acm.org/citation.cfm?id=2540128.2540352`.

Hieu T. Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 79–, New York, NY, USA, 2004. ACM. ISBN 1-58113-838-5. doi: 10.1145/1015330.1015349. URL `http://doi.acm.org/10.1145/1015330.1015349`.

Thang Nguyen, Yuening Hu, and Jordan Boyd-Graber. Anchors regularized: Adding robustness and extensibility to scalable topic-modeling algorithms. In *Proceedings of the Association for Computational Linguistics*, 2014a.

Thang Nguyen, Jordan Boyd-Graber, Jeff Lund, Kevin Seppi, and Eric Ringger. Is your anchor going up or down? Fast and accurate supervised topic models. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2015a.

Viet-An Nguyen, Jordan Boyd-Graber, and Philip Resnik. "I Want to Talk About, Again, My Record On Energy ...": Modeling topic control in conversations using speaker-centric nonparametric topic models. In *Machine Learning Journal*, 2013b.

Viet-An Nguyen, Jordan Boyd-Graber, and Philip Resnik. Lexical and hierarchical topic regression. In *Proceedings of Advances in Neural Information Processing Systems*, 2013c. URL `http://www.cs.colorado.edu/~jbg/docs/2013_shlda.pdf`.

Viet-An Nguyen, Jordan Boyd-Graber, and Philip Resnik. Sometimes average is best: The importance of averaging for prediction using MCMC inference in topic modeling. In *Proceedings of Empirical Methods in Natural Language Processing*, 2014b.

Viet-An Nguyen, Jordan Boyd-Graber, Philip Resnik, and Kristina Miler. Tea party in the house: A hierarchical ideal point topic model and its application to Republican legislators in the 112th Congress. In *Association for Computational Linguistics*, 2015b. URL `http://www.cs.colorado.edu/~jbg/docs/2015_acl_teaparty.pdf`.

Jorge Nocedal and S. Wright. *Numerical Optimization*. 2006.

Peter Norvig. Internet scale data analysis. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 2–2, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0813-7. doi: 10.1145/2020408.2020412. URL `http://doi.acm.org/10.1145/2020408.2020412`.

Edward Orehek and Lauren J. Human. Self-expression on social media: Do tweets present accurate and positive portraits of impulsivity, self-esteem, and attachment style? *Personality and Social Psychology Bulletin*, 43(1):60–70, 2017. doi: 10.1177/0146167216675332. URL `https://doi.org/10.1177/0146167216675332`. PMID: 28903645.

Sebastian Padó and Mirella Lapata. Dependency-based construction of semantic space models. *Comput. Linguist.*, 33(2):161–199, June 2007. ISSN 0891-2017. doi: 10.1162/coli.2007.33.2.161. URL `http://dx.doi.org/10.1162/coli.2007.33.2.161`.

Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '98, pages 159–168, 1998. ISBN 0-89791-996-3.

Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 1901.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

James W. Pennebaker and Martha E. Francis. *Linguistic Inquiry and Word Count*. Lawrence Erlbaum, 1 edition, August 1999. ISBN 156321203X.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL `http://www.aclweb.org/anthology/D14-1162`.

Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Knowledge Discovery and Data Mining*, 2008.

Forough Poursabzi-Sangdeh, Jordan Boyd-Graber, Leah Findlater, and Kevin Seppi. Alto: Active learning with topic overviews for speeding label induction and document labeling. In *Association for Computational Linguistics*, 2016.

Matthew Purver, Konrad Körding, Thomas L. Griffiths, and Joshua Tenenbaum. Unsupervised topic modelling for multi-party spoken discourse. In *Proceedings of the Association for Computational Linguistics*, 2006.

D. Putthividhy, H. T. Attias, and S. S. Nagarajan. Topic regression multi-modal latent dirichlet allocation for image annotation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3408–3415, June 2010. doi: 10.1109/CVPR.2010.5540000.

Sampo Pyysalo, Filip Ginter, Hans Moen, Tapio Salakoski, and Sophia Ananiadou. Distributional semantics resources for biomedical text processing. 2013.

S. Qian, T. Zhang, C. Xu, and J. Shao. Multi-modal event topic model for social event analysis. *IEEE Transactions on Multimedia*, 18(2):233–246, Feb 2016. ISSN 1520-9210. doi: 10.1109/TMM.2015.2510329.

J. R. Quinlan. Induction of decision trees. *Journal of Machine Learning Research*, 1(1):81–106, March 1986. ISSN 0885-6125. doi: 10.1023/A:1022643204877. URL `http://dx.doi.org/10.1023/A:1022643204877`.

Wullianallur Raghupathi and Viju Raghupathi. Big data analytics in healthcare: promise and potential. 2014.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of Empirical Methods in Natural Language Processing*, 2009.

Daniel Ramage, Susan T. Dumais, and Daniel J. Liebling. Characterizing microblogs with topic models. In *International Conference on Weblogs and Social Media*, 2010.

Daniel Ramage, Christopher D. Manning, and Susan Dumais. Partially labeled topic models for interpretable text mining. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 457–465, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0813-7. doi: 10.1145/2020408.2020481. URL `http://doi.acm.org/10.1145/2020408.2020481`.

Steffen Rendle. Factorization machines. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 995–1000, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-4256-0. doi: 10.1109/ICDM.2010.127. URL `http://dx.doi.org/10.1109/ICDM.2010.127`.

Jason Rennie. On l2-norm regularization and the Gaussian prior, 2003.

Philip Resnik and Eric Hardisty. Gibbs sampling for the uninitiated. Technical Report UMIACS-TR-2010-04, University of Maryland, 2010.

Philip Resnik, William Armstrong, Leonardo Claudino, Thang Nguyen, Viet-An Nguyen, and Jordan Boyd-Graber. Beyond lda: Exploring supervised topic modeling for depression-related language in twitter. In *ACL Workshop on Cognitive Modeling and Computational Linguistics, 2015*, 2015.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939778. URL `http://doi.acm.org/10.1145/2939672.2939778`.

Eric K Ringger, Marc Carmen, Robbie Haertel, Kevin D Seppi, Deryle Lonsdale, Peter McClanahan, James L Carroll, and Noel Ellison. Assessing the costs of machine-assisted corpus annotation through a user study. In *LREC*, volume 8, pages 3318–3324, 2008.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1524–1534, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4. URL `http://dl.acm.org/citation.cfm?id=2145432.2145595`.

Christian Robert and George Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer-Verlag, New York, NY, 2004.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241. Springer International Publishing, 2015.

Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. Semeval-2014 task 9: Sentiment analysis in twitter. In *SemEval@COLING*, 2013.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. Semeval-2015 task 10: Sentiment analysis in twitter. In *SemEval@NAACL-HLT*, 2015.

Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, 290:2323–2326, 2000.

Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 441–448, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1. URL `http://dl.acm.org/citation.cfm?id=645530.655646`.

Timothy N. Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. Statistical topic models for multi-label document classification. *Mach. Learn.*, 88(1-2):157–208, July 2012. ISSN 0885-6125. doi: 10.1007/s10994-011-5272-5. URL `http://dx.doi.org/10.1007/s10994-011-5272-5`.

G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975. ISSN 0001-0782. doi: 10.1145/361219.361220. URL `http://doi.acm.org/10.1145/361219.361220`.

Gerard Salton. *Automatic Information Organization and Retrieval.* McGraw Hill Text, 1968. ISBN 0070544859.

Evan Sandhaus. The New York Times annotated corpus, 2008. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp? catalogId=LDC2008T19.

Richard Segal, Ted Markowitz, and William Arnold. Fast uncertainty sampling for labeling large e-mail corpora, 2006.

Jayaram Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

B. Settles. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of Empirical Methods in Natural Language Processing*, 2011a.

Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*, JNLPBA '04, pages 104–107, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1567594.1567618`.

Burr Settles. *Curious Machines: Active Learning with Structured Instances*. PhD thesis, Madison, WI, USA, 2008. AAI3348862.

Burr Settles. Active learning literature survey. *Computer Science Technical Report: Univeristy of Wisconsin, Madison*, 2010.

Burr Settles. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1467–1478, 2011b.

Burr Settles. Active learning (synthesis lectures on artificial intelligence and machine learning). *Long Island, NY: Morgan & Clay Pool*, 2012.

Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 1070–1079, Stroudsburg, PA, USA, 2008a. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1613715.1613855`.

Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1070–1079. Association for Computational Linguistics, 2008b.

Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1289–1296. Curran Associates, Inc., 2008. URL http://papers.nips.cc/paper/3252-multiple-instance-active-learning.pdf.

H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 287–294, New York, NY, USA, 1992a. ACM. ISBN 0-89791-497-X. doi: 10.1145/130385.130417. URL http://doi.acm.org/10.1145/130385.130417.

H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM, 1992b.

Aliaksei Severyn and Alessandro Moschitti. Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 959–962, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3621-5. doi: 10.1145/2766462.2767830. URL http://doi.acm.org/10.1145/2766462.2767830.

C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1): 3–55, January 2001. ISSN 1559-1662. doi: 10.1145/584091.584093. URL http://doi.acm.org/10.1145/584091.584093.

N. Sharma, P. Sharma, D. Irwin, and P. Shenoy. Predicting solar generation from weather forecasts using machine learning. In *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 528–533, Oct 2011. doi: 10.1109/SmartGridComm.2011.6102379.

Xingjian SHI, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 802–810. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5955-convolutional-lstm-network-a-machine-learning-approach-for-precipitation-now
pdf.

Bernard Shizgal. *Spectral Methods in Chemistry and Physics*. Springer-Verlag, 2015.

Alison Smith, Varun Kumar, Jordan Boyd-Graber, Kevin Seppi, and Leah Findlater. Closing the loop: User-centered design and evaluation of a human-in-the-loop topic modeling system. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, pages 293–304, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-4945-1. doi: 10.1145/3172944.3172965. URL http://doi.acm.org/10.1145/3172944.3172965.

Alexander J. Smola and Shravan Narayanamurthy. An architecture for parallel topic models. *International Conference on Very Large Databases*, 3, 2010.

Richard Socher, Yoshua Bengio, and Christopher D. Manning. Deep learning for nlp (without magic). In *Tutorial Abstracts of ACL 2012*, ACL '12, pages 5–5, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2390500.2390505.

David Sontag and Dan Roy. Complexity of inference in latent dirichlet allocation. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1008–1016. Curran Associates, Inc., 2011. URL `http://papers.nips.cc/paper/4232-complexity-of-inference-in-latent-dirichlet-allocation.pdf`.

Mark Steyvers and Thomas Griffiths. Probabilistic topic models. *Handbook of Latent Semantic Analysis*, 2007.

Matt Taddy. On estimation and selection for topic models. In Neil D. Lawrence and Mark Girolami, editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 1184–1193, La Palma, Canary Islands, 21–23 Apr 2012. PMLR. URL `http://proceedings.mlr.press/v22/taddy12.html`.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

M. Teichmann, M. Weber, M. Zöllner, R. Cipolla, and R. Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1013–1020, June 2018. doi: 10.1109/IVS.2018.8500504.

Matus Telgarsky. Dirichlet draws are sparse with high probability. *CoRR*, abs/1301.4917, 2013.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, March 2002. ISSN 1532-4435. doi: 10.1162/153244302760185243. URL `https://doi.org/10.1162/153244302760185243`.

Sho Tsugawa, Yusuke Kikuchi, Fumio Kishino, Kosuke Nakajima, Yuichi Itoh, and Hiroyuki Ohsaki. Recognizing depression from twitter activity. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3187–3196, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3145-6. doi: 10.1145/2702123.2702280. URL `http://doi.acm.org/10.1145/2702123.2702280`.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1858681.1858721`.

Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188, January 2010. ISSN 1076-9757. URL `http://dl.acm.org/citation.cfm?id=1861751.1861756`.

L.J.P. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality reduction: A comparative review, 2008.

S. Van Der Walt, S. C. Colbert, and G. Varoquaux. The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering*, 13(2):22–30, March 2011. ISSN 1521-9615. doi: 10.1109/MCSE.2011.37.

Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996. doi: 10.1137/1038003. URL `http://dx.doi.org/10.1137/1038003`.

Vladimir Vapnik, Esther Levin, and Yann Le Cun. Measuring the vc-dimension of a learning machine. *Neural Comput.*, 6(5):851–876, September 1994. ISSN 0899-7667. doi: 10.1162/neco.1994.6.5.851. URL http://dx.doi.org/10.1162/neco.1994.6.5.851.

Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998. ISBN 0471030031.

Stephen A. Vavasis. On the complexity of nonnegative matrix factorization. *SIAM J. on Optimization*, 20(3): 1364–1377, October 2009. ISSN 1052-6234. doi: 10.1137/070709967. URL http://dx.doi.org/10.1137/070709967.

Seppo Virtanen, Yangqing Jia, Arto Klami, and Trevor Darrell. Factorized multi-modal topic model. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, UAI'12, pages 843–851, Arlington, Virginia, United States, 2012. AUAI Press. ISBN 978-0-9749039-8-9. URL http://dl.acm.org/citation.cfm?id=3020652.3020740.

Stefan Wager, Sida Wang, and Percy Liang. Dropout training as adaptive regularization. In *Proceedings of Advances in Neural Information Processing Systems*, pages 351–359, 2013.

Martin J. Wainwright. Structured regularizers for high-dimensional problems: Statistical and computational issues. Annual Review of Statistics and Its Application, 2014. URL http://www.annualreviews.org/doi/pdf/10.1146/annurev-statistics-022513-115643.

Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

Saint John Walker. Big data: A revolution that will transform how we live, work, and think. *International Journal of Advertising*, 33(1):181–183, 2014. doi: 10.2501/IJA-33-1-181-183.

Hanna Wallach, David Mimno, and Andrew McCallum. Rethinking LDA: Why priors matter. In *Proceedings of Advances in Neural Information Processing Systems*, 2009a.

Hanna M. Wallach. Topic modeling: Beyond bag-of-words. In *Proceedings of the International Conference of Machine Learning*, 2006. ISBN 1-59593-383-2.

Hanna M Wallach. *Structured Topic Models for Language*. PhD thesis, University of Cambridge, 2008.

Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In Leon Bottou and Michael Littman, editors, *Proceedings of the International Conference of Machine Learning*, 2009b.

Chong Wang and David M. Blei. Truncation-free online variational inference for Bayesian nonparametric models. In *Proceedings of Advances in Neural Information Processing Systems*, 2012.

Chong Wang, David Blei, and Li Fei-Fei. Simultaneous image classification and annotation. In *Computer Vision and Pattern Recognition*, 2009a.

Chong Wang, John Paisley, and David Blei. Online variational inference for the hierarchical Dirichlet process. In *Proceedings of Artificial Intelligence and Statistics*, 2011.

Hongning Wang, Minlie Huang, and Xiaoyan Zhu. Extract interaction detection methods from the biological literature. *BMC Bioinformatics*, 10(1):S55, Jan 2009b. ISSN 1471-2105. doi: 10.1186/1471-2105-10-S1-S55. URL https://doi.org/10.1186/1471-2105-10-S1-S55.

Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *Knowledge Discovery and Data Mining*, 2010.

Meng Wang and Xian-Sheng Hua. Active learning in multimedia annotation and retrieval: A survey. *ACM Trans. Intell. Syst. Technol.*, 2(2):10:1–10:21, February 2011. ISSN 2157-6904. doi: 10.1145/1899412.1899414. URL http://doi.acm.org/10.1145/1899412.1899414.

X. Wang, A. McCallum, and X. Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 697–702, Oct 2007a. doi: 10.1109/ICDM.2007.86.

Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Knowledge Discovery and Data Mining*, 2006.

Yi Wang, Hongjie Bai, Matt Stanton, Wen-Yen Chen, and Edward Y. Chang. PLDA: parallel latent Dirichlet allocation for large-scale applications. In *International Conference on Algorithmic Aspects in Information and Management*, 2009c.

Yilun Wang, Wotao Yin, and Yin Zhang. A fast algorithm for image deblurring with total variation regularization, 2007b.

Yining Wang and Jun Zhu. Spectral methods for supervised topic models. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1511–1519. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5517-spectral-methods-for-supervised-topic-models.pdf.

Xing Wei and W. Bruce Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 178–185, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. doi: 10.1145/1148170.1148204. URL http://doi.acm.org/10.1145/1148170.1148204.

F. M. F. Wong, C. W. Tan, S. Sen, and M. Chiang. Quantifying political leaning from tweets, retweets, and retweeters. *IEEE Transactions on Knowledge and Data Engineering*, 28(8):2158–2172, Aug 2016. ISSN 1041-4347. doi: 10.1109/TKDE.2016.2553667.

J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, Feb 2009. ISSN 0162-8828. doi: 10.1109/TPAMI.2008.79.

Xindong Wu, Xingquan Zhu, Gong-Qing Wu, and Wei Ding. Data mining with big data. 2013.

Pengtao Xie and Eric P. Xing. Integrating document clustering and topic modeling. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, pages 694–703, Arlington, Virginia, United States, 2013. AUAI Press. URL http://dl.acm.org/citation.cfm?id=3023638.3023709.

Pengtao Xie, Diyi Yang, and Eric Xing. Incorporating word correlation knowledge into topic modeling. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 725–734. Association for Computational Linguistics, 2015a. doi: 10.3115/v1/N15-1074. URL http://aclweb.org/anthology/N15-1074.

Pengtao Xie, Diyi Yang, and Eric P. Xing. Incorporating word correlation knowledge into topic modeling. In *In Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015b.

Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. Rc-net: A general framework for incorporating knowledge into word representations. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 1219–1228, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2598-1. doi: 10.1145/2661829.2662038. URL http://doi.acm.org/10.1145/2661829.2662038.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France, 07–09 Jul 2015. PMLR. URL `http://proceedings.mlr.press/v37/xuc15.html`.

Rui Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3): 645–678, May 2005. ISSN 1045-9227. doi: 10.1109/TNN.2005.845141.

Zuobing Xu, Ram Akella, and Yi Zhang. Incorporating diversity and density in active learning for relevance feedback. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, pages 246–257, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-71494-1. URL `http://dl.acm.org/citation.cfm?id=1763653.1763684`.

Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, pages 181–213, 2015. ISSN 0219-3116. URL `https://doi.org/10.1007/s10115-013-0693-z`.

Tze-I Yang, Andrew Torget, and Rada Mihalcea. Topic modeling on historical newspapers. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, June 2011.

Limin Yao, David Mimno, and Andrew McCallum. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 937–946, New York, NY, USA, 2009a. ACM. ISBN 978-1-60558-495-9. doi: 10.1145/1557019.1557121. URL `http://doi.acm.org/10.1145/1557019.1557121`.

Limin Yao, David Mimno, and Andrew McCallum. Efficient methods for topic model inference on streaming document collections. In *Knowledge Discovery and Data Mining*, 2009b. ISBN 978-1-60558-495-9.

David Newman Yee Whye Teh and Max Welling. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. *Proceedings of Advances in Neural Information Processing Systems*, 2006.

Dani Yogatama and Noah Smith. Linguistic structured sparsity in text categorization. 2014.

Dani Yogatama, Manaal Faruqui, Chris Dyer, and Noah Smith. Learning word representations with hierarchical sparse coding. In David Blei and Francis Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 87–96. JMLR Workshop and Conference Proceedings, 2015. URL `http://jmlr.org/proceedings/papers/v37/yogatama15.pdf`.

Michelle Yuan, Benjamin Van Durme, and Jordan L Ying. Multilingual anchoring: Interactive topic modeling and alignment across languages. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8667–8677. Curran Associates, Inc., 2018. URL `http://papers.nips.cc/paper/8083-multilingual-anchoring-interactive-topic-modeling-and-alignment-across-langu pdf`.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 68:49–67, 2006.

Omar F. Zaidan, Jason Eisner, and Christine Piatko. Machine learning with annotator rationales to reduce annotation cost. In *Proceedings of the NIPS*2008 Workshop on Cost Sensitive Learning*, Whistler, BC, 2008. 10 pages.

Ke Zhai and Jordan Boyd-Graber. Online latent Dirichlet allocation with infinite vocabulary. In *Proceedings of the International Conference of Machine Learning*, 2013.

Ke Zhai, Jordan Boyd-Graber, Nima Asadi, and Mohamad Alkhouja. Mr. LDA: A flexible large scale topic modeling package using variational inference in mapreduce. In *Proceedings of the World Wide Web Conference*, 2012.

Ke Zhai, Jordan Boyd-Graber, and Shay B. Cohen. Online adaptor grammars with hybrid inference. 2014.

Peng Zhao, Guilherme Rocha, and Bin Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *Ann. Statist*, 2009.

Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 487–495. Curran Associates, Inc., 2014. URL `http://papers.nips.cc/paper/5349-learning-deep-features-for-scene-recognition-using-places-database.pdf`.

Jun Zhu, Amr Ahmed, and Eric P. Xing. MedLDA: maximum margin supervised topic models for regression and classification. In *Proceedings of the International Conference of Machine Learning*, 2009a. ISBN 978-1-60558-516-1. doi: http://doi.acm.org/10.1145/1553374.1553535.

Jun Zhu, Ning Chen, Hugh Perkins, and Bo Zhang. Gibbs max-margin topic models with fast sampling algorithms. In *Proceedings of the International Conference of Machine Learning*, 2013.

Xiaojin Zhu, Andrew B. Goldberg, Ronald Brachman, and Thomas Dietterich. *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers, 2009b. ISBN 1598295470, 9781598295474.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. 2005.