Thesis

Beyond Keyword Search: Representations and Models for Personalization

Khalid El-Arini

CMU-CS-13-102

January 29, 2013

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Carlos Guestrin, Chair
Zoubin Ghahramani
Tom Mitchell
Noah Smith
Thorsten Joachims, Cornell University

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was partially supported by the Office of Naval Research under MURI N000141010934, MURI N000140710747, YIP N000140810752 and PECASE N000141010672, the National Science Foundation under CAREER IIS0644225, NeTS-NOSS CNS0625518 and NeTS-SCAN CNS0721591, and by the Army Research Office under MURI W911NF0710287 and W911NF0810242.

Keywords: personalization, recommendation, transparency, user studies, social networks, Twitter, document representation, content analysis, topic modeling, graphical models, sparsity, machine learning, information retrieval

Abstract

We live in an era of information overload. From online news to online shopping to scholarly research, we are inundated with a torrent of information on a daily basis. With our limited time, money and attention, we often struggle to extract actionable knowledge from this deluge of data. A common approach for addressing this challenge is *personalization*, where results are automatically filtered to match the tastes and preferences of individual users. While showing promise, modern systems and algorithms for personalization face their own set of challenges, both technical and social in nature. On the technical side, these include the well-documented "cold start" problem, redundant result sets and an inability to move beyond simple user interactions, such as keyword queries and star ratings. From a social standpoint, studies have shown that most Americans have negative opinions of personalization, primarily due to privacy concerns.

In this thesis, we address these challenges by introducing *interactive concept coverage*, a general framework for personalization that incentivizes diversity, and applies in both queryless settings as well as settings requiring complex and rich user interactions. This framework involves framing personalized recommendation as a probabilistic budgeted max-cover problem, where each item to be recommended is defined to *probabilistically cover* one or more *concepts*. From user interaction, we learn weights on concepts and affinities for items, such that solving the resulting optimization problem results in personalized, diverse recommendations. Theoretical properties of our framework guarantee efficient, near-optimal solutions to our objective function, and no-regret learning of user preferences.

We show that, by using the interactive concept coverage methodology, we are able to significantly outperform both state-of-the-art algorithms and industrial market leaders on two important personalization domains: news recommendation and scientific literature discovery. Empirical evaluations—including live user studies—demonstrate that our approach produces more diverse, more relevant and more trustworthy results than leading competitors, with minimal burden on the user. Finally, we show that we can directly use our framework to introduce a level of *transparency* to personalization that gives users the opportunity to understand and directly interpret (and correct) how the system views them.

By successfully addressing many of the social and technical challenges of personalization, we believe the work in this thesis takes an important step in ameliorating problems of information overload.

Contents

Acknowledgments

A wise man once said, “If you were successful, somebody along the line gave you some help....you didn’t build that.” As I finish writing my thesis—after years of support from mentors, colleagues, family and friends—it is hard to imagine any human endeavor where this statement rings more true.

I must start by thanking my advisor, Carlos Guestrin. Carlos is a passionate teacher who drives his students to excel. He has a brilliant, creative mind, and any success I have achieved over the course of my doctoral studies would not have been possible without his steady guidance and tutelage. Most of all, I am grateful for the friendship and camaraderie Carlos developed with his students and encouraged within his group.

I am also indebted to many other faculty members—at Carnegie Mellon and elsewhere—who played important roles in my graduate studies. First and foremost, Tom Mitchell, as my current committee member and former advisor, was always ready with insightful advice and a cheerful smile. My journey as a doctoral student took a long and winding path, and Tom is largely responsible for making sure I reached this point. Rounding out my distinguished committee, Zoubin Ghahramani, Thorsten Joachims and Noah Smith each imparted wisdom through several research discussions over the last two years, and I am thankful for their advice and ideas. Beyond my committee, Geoff Gordon deserves special mention and gratitude for his role as co-director of the Select Lab. Learning seems to happen simply by sitting in the same room as Geoff, who is one of the smartest people I have had the honor of getting to know. Other faculty who have had lasting impact on me throughout my time as a graduate student include David Blei, Emily Fox, Arthur Gretton, Niki Kittur, Andrew Moore and Alex Smola. Special thanks also goes to Todd Mowry, who was my undergraduate senior thesis advisor, helping me get started with research at the very beginning.

I have no single-author papers. Rather, I have been fortunate to collaborate with an exciting group of researchers, and I hope I have given to them just a fraction of the inspiration, insight and knowledge that they have given me. The initial work of this thesis was in collaboration with Gaurav Veda and Dafna Shahaf, and it is doubtful that the simultaneous intensity and hilarity of the meetings the three of us had with Carlos can ever be surpassed. Whether it was bouncing ideas off of each other, coding late into the night, negotiating Middle East peace, or walking along the Seine, that collaboration was one of the most treasured aspects of my time as a graduate student. Soon after, Yisong Yue joined our lab as a postdoctoral fellow, and working with Yisong on principled machine learning methods for information retrieval has been a truly educational experience. He has also been invaluable in helping me design and conduct user studies, which have been an integral part of my thesis work. Over the last two years, Emily Fox has played an important role in my research, and I was very fortunate to spend some time collaborating with her at both the University of Pennsylvania and the University of Washington, in addition to the frequent visits she made to Carnegie Mellon. Emily has taught me much of what I know about Bayesian inference and nonparametric methods, and it has been a distinct pleasure to work with her on two exciting projects. More recently, I have had the honor of working with Min Xu, who has been my trusted guide and mentor as I

dipped my toes in frequentist waters over the last year. Deep gratitude goes to Brendan O'Connor, who helped me tremendously with the final portion of my research by facilitating my access to the necessary Twitter data. Finally, I must thank my collaborators at Microsoft Research Cambridge, who made my summer in England a fruitful one: Ralf Herbrich, Ulrich Paquet, Jurgen Van Gael and Blaise Agüera y Arcas.

It takes something quite special for someone with an office in the fancy Gates Center to wax nostalgic about a windowless basement room in old Wean Hall, but to me (and many of my lab mates), the Select Lab fits the bill. It is uncommon for graduate students to belong to a lab as close-knit as ours, and one of the few tragedies of moving to a new building was losing our common workspace. The grilling we each received in our lab meetings, practice talks and reading groups prepared us for anything a hostile conference audience might bring, and undoubtedly made us better scientists. I am thankful for the friendships and collaborations with members of the lab, past and present: Danny Bickson, Byron Boots, Joseph Bradley, Anton Chechetka, Carlton Downey, Miro Dudík, Stano Funiak, Joey Gonzalez, Arthur Gretton, Jay Gu, Ahmed Hefny, Sue Ann Hong, Jonathan Huang, Adona Iosif, Shiva Kaul, Andreas Krause, Aapo Kyrölä, Wooyoung Lee, Yucheng Low, Austin McDonald, Ram Ravichandran, Sajid Siddiqi, Dafna Shahaf, Ajit Singh, Gaurav Veda, Yisong Yue, Erik Zawadzki and Brian Ziebart. Thanks as well to our summer interns Samuel Hopkins and Nara Kasbergen.

My officemates Kevin Killourhy, Mary McGlohon, Anton Chechetka, Julian Shun, Mukesh Agrawal, Kyung-Ah Sohn and Nicole Rafidi were great companions through thick and thin. (Special apologies are due to Mary for all the times we knocked on our door with her advisor's signature knock.)

Our lab and department would not run without the herculean efforts of Michelle Martin, Diane Stidle and Deb Cavlovich, all three of which I thank wholeheartedly for all that they do. Mark Stehlik, my tireless undergraduate academic advisor and friend, deserves a special round of thanks as well.

I have been blessed with many great friendships at Carnegie Mellon and in Pittsburgh, and if I started to list all of my friends one by one, describing how much each one means to me, this acknowledgments section would soon be at least twice as long. Instead, I offer a heartfelt thank you to all of you; you know who you are, and without your support, this thesis would not have happened. Thanks for helping me grow intellectually and spiritually over the last several years, and being sounding boards for my thoughts, worries and dreams. (And thanks for coming to our wedding in Cairo, despite being two blocks away from a turbulent Tahrir.) I am especially grateful to Sue Ann Hong, Gaurav Veda, Mary McGlohon, Sajid Siddiqi, Nada Quraishi, Jonathan Huang, Bri-Mathias Hodge, Gilbert Dussek and Bobby Oberreuter, for being there for me when times were tough.

My parents, Bakry and Zeinab El-Arini, have always been most generous with their time, love and support, and anything I have accomplished is a testament to them. They are my role models and lifelong mentors, teaching me the values of hard work and selflessness. I am thankful to them for everything in my life. I am also grateful for the irreplaceable support from my younger brother, Ashraf; when times got tough at graduate school, it was enough to know that I could talk to him about any number of topics to put my mind at ease, from our beloved Washington sports teams to details of climate policy. My extended family in Egypt and beyond has always been close to my heart, despite the geographic distance. Special thanks go to my uncles Omar, Sam and Farid, and my Aunt Silvia, who along with my father, would tell me stories of their doctorates to buoy my spirits and keep me going. My cousin Mai, her husband Ahmad and their beautiful children moved to Pittsburgh for two years during my studies, and I cannot thank them enough for making me feel at home in my frequent visits to their house. I also must thank the Kosbas—particularly, Uncle Taha, Aunt Khadiga and Reem— for their unwavering support and love.

Most of all, I am grateful to May Kosba, my wife, best friend and purveyor of happiness. God has provided me with many blessings in life, but none are dearer to me than your love and support. I truly could not have finished this thesis without you being there for me, and I am looking forward to a lifetime together filled with happy adventures.

Chapter 1

Introduction

> "The scarce resource in the age of digital journalism is not high-quality content, but attention."
>
> *—Ethan Zuckerman, Berkman Center for Internet and Society at Harvard University, 2010*

As early as 1755, the French philosopher Denis Diderot presciently forewarned that there would come a day when "it will be almost as convenient to search for some bit of truth concealed in nature as it will be to find it hidden away in an immense multitude of bound volumes," [Diderot, 1755]. Today, Diderot's "immense multitude" is present in nearly every domain, from online news and online shopping to social networks and scientific research.

Web users trying to gauge public opinion or learn about current events face a torrent of information from tens of millions of sources worldwide. For example, the blog indexing service Spinn3r retrieves over one million news articles, blog posts and social media updates every hour.[1] At the time of writing, there were nearly fifty billion indexed pages on the World Wide Web,[2] over 72 hours of video uploaded per minute to YouTube,[3] and over 400 million tweets a day on the microblogging site Twitter.[4] Similarly, for generations, scientists have built upon the published work of their predecessors and contemporaries in order to make new discoveries. However, with tens of millions of articles published in tens of thousands of journals and conferences,[5] researchers face an acute difficulty in sifting through related literature.

Today, most of these information overload problems are addressed using keyword search. Google and other commercial search engines have been successful at efficiently providing highly relevant content in response to user queries consisting of short strings of words. However, many common information retrieval tasks do not fit into this traditional keyword search paradigm. Notably, some information needs are *not naturally representable as queries*. For example, reading the day's news is inherently a queryless process, as is receiving the most relevant updates from friends on a social network. In other cases, an information need may have a natural query, but is *too complex to be expressed as keywords*. An instructive example is attempting to use Google Scholar to discover related scientific literature: It is easy to find a specific author or paper, but there is no way to specify a particular research question that returns meaningful, helpful and non-trivial results.

[1] `http://www.spinn3r.com`
[2] `http://www.worldwidewebsize.com`
[3] `http://www.youtube.com`
[4] `http://www.twitter.com`
[5] Statistics from Thomson Reuters: `http://wokinfo.com/about/whatitis/`

These shortcomings of keyword search are presently dealt with in three primary ways:

1. The problems with keyword search are sometimes simply *ignored* by the service provider, and users are left to make the best of the situation on their own. They may try (and try again) to express their complex information needs within the confines of a small text box, and once that inevitably fails, will be forced to resort to inefficient browsing behavior. For example, with Google Scholar, users who cannot find what they are looking for using keywords often have no recourse but to scan through proceedings of relevant conferences or follow long citation trails, with the hope of finding a useful article.

2. In queryless settings, a simplifying assumption is often made that all users will have the same information need, thus resulting in a *single common view* to all users. For example, many online newspapers subscribe to such a notion.

3. In domains where it is clear that different users are unlikely to have the same tastes (e.g., music or films), service providers may provide *personalized recommendations* based on simple user interactions, to replace or supplement keyword search. Some famous examples include Netflix's movie recommendation and Amazon's product recommendation.

While the first approach can lead to bad user experiences, and the second is overly simplistic in many cases, the third—*personalization*—has the potential, when executed correctly, to quickly satisfy a user's specific information need. However, personalization is not without its own problems; the subject of this thesis will be to address many of these problems in such a way that we can demonstrably show how personalization can succeed in solving complex information retrieval tasks in settings where traditional keyword search alone is inadequate.

1.1 Personalization and its Discontents

Personalization, particularly in the queryless setting, can help efficiently direct the user to relevant content. However, several caveats exist that can hold back the performance of a personalization system, and must be addressed if a personalization approach is to be effective:

1. *The cold start problem* exists when a personalization system does not have enough information to make a recommendation for a new user or a new item. For example, a pure collaborative filtering system for news recommendation will have trouble recommending a newly written article, because it depends solely on past ratings assigned to the article by other users.

2. The personalization of content exists across *multiple dimensions*, which may be completely orthogonal to each other. For example, finding out that a user is interested in reading about French cuisine is qualitatively different than learning that the user trusts *The Guardian* more than *The Washington Post*. User models should be rich enough to encode such different forms of user preferences.

3. Likes, clicks and star ratings are useful user interactions, but are not the limit of what is possible or useful. *Richer user interactions* can allow for more powerful personalization by freeing the user to be more expressive in defining his or her information need.

4. As the quote at the beginning of this chapter alludes, the number of high quality, relevant articles (or movies or books) exceeds that which can be processed by a single user at one time. Human attention is a scarce resource that must be modeled when designing a personalization system, and

thus *redundancy* of results must be strictly avoided. If a user skips a news article because he finds it to be redundant to one that he was previously shown, it is a waste of his attention and a lost opportunity to show him another relevant article.

5. Users are often unaware that personalization is taking place, and when they are aware, they often find it disconcerting. In a 2012 study by the Pew Internet and American Life Project,[6] 65% of respondents described personalized search as a "bad thing," while 73% of respondents described personalization as an "invasion of privacy." Moreover, users sometimes worry that their results are being *overpersonalized*, and they are stuck in the so-called Filter Bubble [Pariser, 2011], confining them to an echo chamber where they only see a narrow view of the world.

In this thesis, we introduce the general framework of *interactive concept coverage* in order to address each one of these problems.

1.2 Interactive Concept Coverage

Consider a setting where we are tasked with recommending relevant news articles to a user, personalized to her tastes. News is a dynamic domain with time-sensitive content, and thus we cannot afford to wait until many other users have read an article before making our recommendation; in other words, we cannot wait for the cold start problem to go away on its own, or else we would be recommending old news. Rather, we must use the content of the articles to decide which documents to present to the user.

In order for a computer to reason about a document in this manner, a representation must be chosen for the document's contents. One of the simplest and most common representations for document modeling is known as the "bag of words" model, where the order of the words in a document is ignored, and the content of a document is simply represented as a list of words and their frequencies [Salton et al., 1975]. Some more sophisticated representations count n-grams, named entities, noun phrases and richer syntactic structures, while others represent a document as being generated from latent factors known as topics [Blei and Lafferty, 2009].

What these representations have in common is that they distill the contents of a document collection into fundamental *concepts* that represent the atomic units of information to be reasoned about. The simple approach we develop in this thesis, which we call *interactive concept coverage*, utilizes such a concept representation of a document collection in order to provide diverse, personalized recommendations.

At a high level, our approach entails the following six steps:

1. We first define a *weighted concept representation* of the content domain from which we wish to recommend. In our news recommendation example, we might represent the news of the day as a collection of named entities and noun phrases. For instance, last October, there may have been high weight on concepts representing Barack Obama and Mitt Romney, due to the American presidential elections, as well as significant weight on Syria and Hurricane Sandy. This weighting should take into account the general importance of a concept in addition to anything we know about the particular user's tastes and preferences.

[6] `http://www.pewinternet.org/Reports/2012/Search-Engine-Use-2012/Summary-of-findings.aspx`

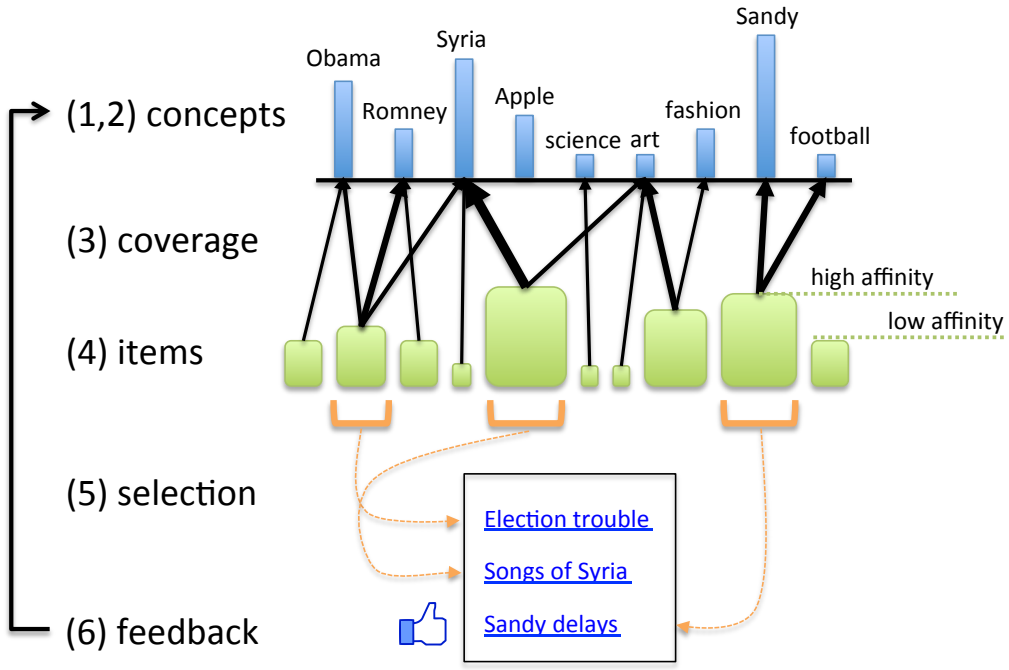


Figure 1.1: This figure summarizes our *interactive concept coverage* approach, from the point of view of a news recommendation system. Numbers in parentheses correspond to the six steps listed below. In this diagram, we have nine concepts weighted by importance, indicated by blue bars. The green rectangles correspond to news articles, and the thickness of the black arrows emanating from them indicates how much a particular article covers a given concept. For example, the leftmost article covers only the "Obama" concept, and does so only slightly. The different sizes of the green rectangles allude to the personalized affinity the user might have for the articles' non-content features, such as preferring *The New York Times* to Fox News. A set of diverse articles, indicated in orange, is selected and presented to the user, who gives feedback which is then used to modify the concept weighting for future visits by the user.

2. In the cases where our information retrieval task includes a query, we *solicit the user for a query* of some sort, and potentially use this information to modify the concept representation. The query does not have to be limited to a traditional keyword string, but could include richer forms of interaction, such as specifying a query set of documents, highlighting a portion of a webpage, or specifying a location on a map. We skip this step in a queryless setting such as news recommendation.

3. We define a *coverage function* over the individual items to be recommended that describes how much each item *covers* each of the concepts. For example, an article about sports is unlikely to cover the "Syria" concept, whereas an article about the Arab Spring should cover the "Syria" concept strongly.

4. We define a *personalized affinity function* over the non-content features of the items to be recommended, allowing us to represent preferences unrelated to content, such as "I trust the New York Times" or "I prefer articles that have many votes on Reddit." These preferences might be directly specified by the user, or otherwise learned from user feedback.

5. We *return a set of items*—up to a budget constraint—that collectively covers the most important concepts while penalizing redundancy and incentivizing items with high affinity features. In news recommendation, the budget constraint comes from the fact that a user often does not have much

time (or screen pixels) to view more than a few articles at once.

6. We record *feedback from the user* that allows us to update the concept weights and affinity functions, such that the next round of personalization for this user is more accurate. In our running example, this might involve traditional feedback such as a user clicking "like" or "dislike" on a particular news article.

These steps are summarized in Figure 1.1.

1.3 Thesis Statement and Contributions

The core of this thesis research revolves around the following statement:

> In a variety of information overload settings, the Interactive Concept Coverage framework produces personal recommendations that are highly relevant, diverse and transparent, incorporating complex user preferences through rich user interaction.

We evaluate this thesis statement by showing that we can successfully address all of the aforementioned pitfalls of personalization by following the interactive concept coverage methodology in both queryless and complex query settings.

The specific contributions of this thesis are as follows:

- We introduce the *interactive concept coverage* framework as a general methodology for producing diverse, personalized results to information retrieval problems.
- We demonstrate the interactive concept coverage framework in a queryless news recommendation setting with simple like/dislike interactions. Here, we introduce notions of document coverage functions and set coverage functions that encode a natural diminishing returns property that incentivizes diversity in the result set. We also introduce a simple online learning algorithm that provides regret guarantees on learning user preferences over concepts from limited feedback. Experimental results show that our approach beats state-of-the-art alternatives. (Chapter 3)
- We demonstrate the interactive concept coverage framework in a setting with complex queries: discovering relevant scientific literature. We show an example of how such queries can be encoded using an augmented concept representation. We also provide algorithms for computing a personalized affinity function over research papers based on a user's trust preferences. We show experimentally that our approach produces more relevant, more diverse, and more trustworthy results than state-of-the-art competitors, including Google Scholar. (Chapter 4)
- We address the problem of *overpersonalization* by defining a transparent user model based on the self-described attributes of Twitter users, which we call *badges*. We introduce a probabilistic model that allows us to learn a user's badges from his or her Twitter activity, devise an inference algorithm for the model, and show that we can successfully beat state-of-the-art alternatives at predicting a user's attributes from his or her actions. Moreover, we emphasize the interpretability of these learned user features. (Chapter 5)
- We investigate the problem of *document representation* by studying multiple approaches for distilling the content of a document collection into atomic concepts. We devise an approach based on *badges*

that allows us to represent documents by attributes of their likely readers, and experimentally show that such a concept representation is both interpretable and effective at personalization. (Chapter 6)

1.4 Outline

Chapter 2 introduces common terminology and background material that will help the reader understand the rest of the thesis. Chapters 3-6 provide details on the contributions described above, each containing algorithmic details, results and extended related work. Chapter 7 details conclusions and insights gained from this thesis work, while Chapter 8 introduces ideas for future work.

Chapter 2

Background

2.1 Probabilistic Graphical Models

This section is intended as a brief overview of a rich and complex subfield of machine learning. Combining elements of both graph theory and probability theory, *probabilistic graphical models* have become a fixture of modern statistical methods and real-world applications, from computer vision to computational biology. Here, we provide a high-level introduction that will allow the reader to better understand the model presented in Chapter 5. However, for more details, we direct the interested reader to one of many thorough survey papers and books [Jensen, 2001, Jordan, 2004, Koller and Friedman, 2009, Lauritzen, 1996, Pearl, 1988, Wainwright and Jordan, 2008].

As a motivating example, let us consider a collection of N political columnists. We are interested in modeling the political leanings of each pundit, so that we can show users articles from different viewpoints. We assume a simplistic model, where each columnist i is represented by a binary random variable $X_i \in \{\text{left}, \text{right}\}$, indicating whether he is left-leaning or right-leaning.[1] By analyzing the articles that these columnists write, we might try to infer how each one leans politically.

Given such a setup, we can ask: what is the most likely configuration of the biases of the political punditry? In order to answer such a question, we need access to the full joint distribution over the variables $\{X_i\}$. Naïvely, to estimate a discrete distribution over N binary variables, we would need to learn $2^N - 1$ parameters, one for each possible configuration (taking into account that a discrete probability distribution must sum to 1). This exponential dependence on N requires an inordinately large amount of training data for us to accurately estimate all of the parameters.

However, what if we assumed that, when it comes to political leanings, the columnists are fully *independent* of each other?[2] Namely, the fact that one columnist, Alice, leans left plays no role in determining the political bias of any another columnist, e.g., Bob. In such a scenario, the joint probability can be factorized over the N independent columnists, leaving us only N parameters to estimate.

Of course, columnists are not necessarily (marginally) independent of each other; some might work for the same newspaper, others might live in the same city, while others yet might have similar childhood or educational experiences, leading them to have similar outlooks on life. For example, if we assume

[1] In our example, we assume political pundits are never unbiased.

[2] For all disjoint subsets $\mathbf{Y}$ and $\mathbf{Z}$ of $\{X_i\}$, we assume $\mathbf{Y}$ and $\mathbf{Z}$ are independent.

a pundit's political leanings are solely a result of where she lives, then after observing her location L_i, her political leaning X_i is *conditionally independent* of X_j for all $j \neq i$. The machinery of probabilistic graphical models allows us to cleanly model such *conditional independence* relationships, leading to a compact representation of complex probability distributions.

2.1.1 Representation

As the name indicates, a probabilistic graphical model is a graph-based representation of a probability distribution. We define a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to have a set of vertices, $\mathcal{V}$, and a set of edges, $\mathcal{E}$. An edge $(s, t) \in \mathcal{E}$ connects two vertices in $\mathcal{V}$. Probabilistic graphical models have different semantics based on whether the edges in the graph are directed or undirected. In this thesis, we exclusively deal with directed graphical models—commonly referred to as *Bayesian networks* or *Bayes nets*—and so in this background section, we will we will only discuss the semantics for this case. As such, we assume each edge $(s, t) \in \mathcal{E}$ is directional, pointing from a source vertex $s \in \mathcal{V}$ to a destination vertex $t \in \mathcal{V}$.

Definition 2.1.1 (Bayesian network)**.** *A* Bayesian network *is a triplet $(\mathbf{X}, \mathcal{G}, \mathcal{P})$, where $\mathbf{X} = \{X_i\}$ is a set of random variables, $\mathcal{G} = (\mathbf{X}, \mathcal{E})$ is a directed, acyclic graph (DAG) over the variables $\{X_i\}$, and $\mathcal{P}$ is a set of conditional probability distributions, one per vertex. Specifically, for each random variable X_i, we have the conditional probability distribution of X_i given its parents,*[3] *$P(X_i|Pa(X_i))$, in the set $\mathcal{P}$.*

Given this definition, we can ask the following questions to help us understand the semantics of a Bayesian network:

Q1. How does the graph $\mathcal{G}$ encode conditional independence assumptions among the random variables $\{X_i\}$?

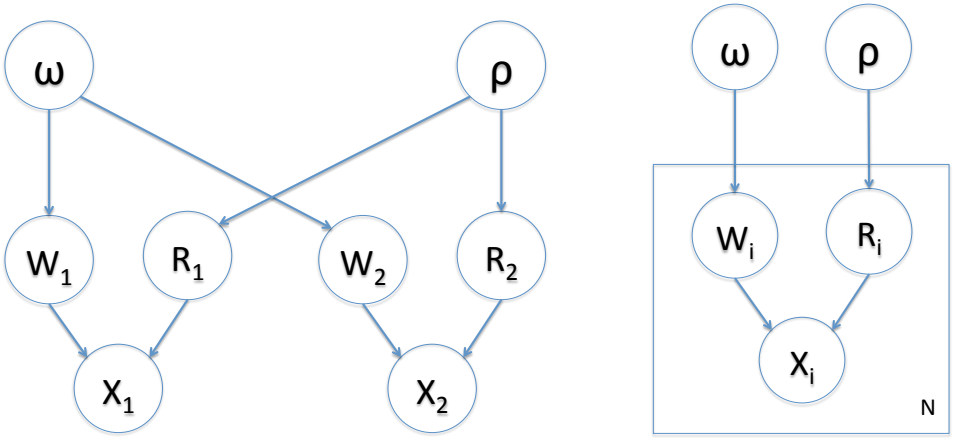
To answer this question, we first define the *Local Markov Property* of a Bayesian network:

Definition 2.1.2 (Local Markov Property)**.** *Given a Bayesian network over the variables $\mathbf{X}$, represented by the graph $\mathcal{G}$, a variable X_i is conditionally independent of its non-descendants given its parents (and only its parents).*

To illustrate this property, let us return to our example of pundits and their political leanings. Figure 2.1a shows a Bayesian network representation of a slightly more complicated model over the political leanings of two pundits. Here, X_i, as before, indicates whether the pundit leans left or right. We assume that the political leaning of a pundit is based on two factors: his wealth, represented by the random variable W_i, and the identity of his childhood role model, represented by R_i. For example, a wealthy pundit who grew up idolizing Ronald Reagan may be more likely to lean right, politically. This is represented in the figure by directed edges to X_i from W_i and R_i. We also assume global prior distributions over the wealth of pundits (parameterized by ω) and their role models (parameterized by ρ). The local Markov property indicates that X_1, given its parents W_1 and R_1, is conditionally independent of its non-descendants, W_2, R_2, X_2, λ and ω.

It is natural to wonder whether the local Markov property implies any other conditional independence relations. For example, from inspecting the graph in Figure 2.1a, can we decide whether W_1 and W_2 are marginally independent? What about W_1 and R_1? Are W_1 and R_1 conditionally independent given X_1?

[3] We define the parents of X_i, $Pa(X_i)$, in the graph $\mathcal{G}$ to be the set of vertices in $\mathcal{G}$ that have a directed edge terminating in X_i. Specifically, $Pa(X_i) = \{X_j : (X_j, X_i) \in \mathcal{E}\}$.



(a) (b)

Figure 2.1: **(a)** An example Bayesian network modeling the political leanings of two pundits. **(b)** A plate diagram extending the model of (a) to N pundits.

More generally, given a graph $\mathcal{G}$, how do we decide whether a variable X is conditionally independent of a variable Y given a set of observed evidence variables $\mathbf{E}$?[4] In his seminal work on graphical models, Pearl showed that questions like these can in fact be answered by using the concept of *d-separation* [Pearl, 1988]. Following Schachter [1998], we explain d-separation by making a simple analogy to a ball traveling along the edges of the graph $\mathcal{G}$. Specifically, if we want to determine if, given evidence $\mathbf{E}$, X is conditionally independent of Y, we can place a ball at X, and see if the ball can successfully travel to Y without being blocked. If this is the case, then we say there is an active path between X and Y, given $\mathbf{E}$, and the two variables are *not* conditionally independent given the evidence. If there is no active path, we say the two variables are d-separated, and, hence, conditionally independent.

Figure 2.2 depicts the rules governing the valid paths a ball can take through the graph. In our example, we can use these rules to learn that W_1 and R_1 are marginally independent (there is no active path between them), but that if we observe X_1, then W_1 and R_1 become statistically dependent through their common effect (also known as a *v-structure*).

Q2. Given the conditional distributions in $\mathcal{P}$, how do we write the joint distribution over all the random variables in $\{X_i\}$?
Given a Bayesian network over $\{X_i\}$ specified by the graph $\mathcal{G}$ and the conditional distributions $\mathcal{P}$, the joint distribution over the entire set of random variables can be written as follows:

$$P(X_1, X_2, \ldots, X_N) = \prod_{i=1}^{N} P(X_i | Pa(X_i)). \tag{2.1}$$

Q3. Can any joint probability distribution over a set of variables $\{X_i\}$ be represented as a Bayesian network?
Given a graph $\mathcal{G}$ and a probability distribution P, both over the set of variables $\{X_i\}$, the Representation Theorem of Bayesian networks (cf. Theorem 3.1 and 3.2 of [Koller and Friedman, 2009]) tells us that:

[4] We write this conditional independence statement as $X \perp Y | \mathbf{E}$.

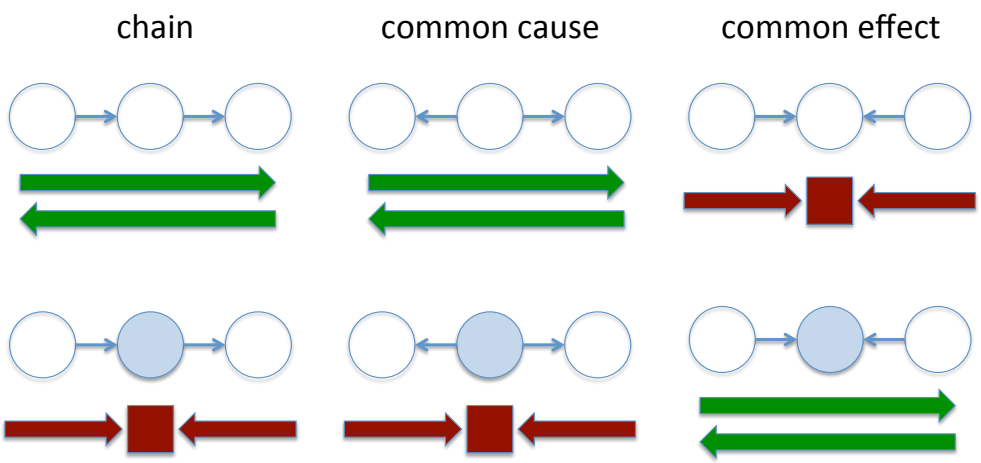


Figure 2.2: In order to determine whether two variables X and Y are d-separated given a set of evidence variables $\mathbf{E}$, we can employ the *Bayes Ball* algorithm [Schachter, 1998]. In this algorithm, we decide whether a hypothetical ball can travel between the nodes X and Y in the graph, without being blocked. This figure shows the rules governing whether the ball can pass through a node in the graph or not. These rules depend on the directions of the edges going into or out of the node, as well as whether the node is observed or not (i.e., whether the node is in the evidence set $\mathbf{E}$). Following standard convention, we depict an evidence node by shading it. Here, green arrows depict cases where the ball can successfully travel through the node, whereas the red blockade corresponds to cases where the ball is blocked.

1. If the conditional independence relations encoded in a Bayesian network on $\mathcal{G}$ are a subset of the conditional independence relations that exist in P, then P can be factorized over the graph $\mathcal{G}$ (as shown in Eq. (2.1)); and,

2. Any probability distribution P that decomposes over $\mathcal{G}$ (as shown in Eq. (2.1)) can be represented as a Bayesian network defined by $\mathcal{G}$, such that every independence assumption implied by the local Markov property of the graph also exists in the distribution P.

A corollary of this theorem is that we can represent any distribution P over $\{X_i\}$ by a directed, acyclic clique over the variables $\{X_i\}$. The reason for this is that a fully connected graph does not encode any independence assumptions whatsoever, and thus the null set of independence assumptions that it encodes is trivially a subset of any conditional independence assumptions found in an arbitrary distribution P. Of course, such a representation would not be compact, as we would have to define the conditional probability of each node given its large set of parents, which returns us to the problem that motivated our discussion of graphical models in the first place.

Before we move on to the next section, let us return to our example on the political biases of pundits, and consider once more Figure 2.1a. In this figure, we are modeling the political leanings of two pundits, represented by variables X_1 and X_2. This is a straightforward diagram, since it is easy to visualize all the relationships connecting these two sets of variables. However, how do we represent the hundreds of thousands of writers who opine on political topics every day? In Figure 2.1b, we use a *plate notation*, where we take advantage of the fact that each pundit will be modeled in exactly the same way (with variables X_i, W_i and R_i). The rectangle around this triplet of variables has an N in the corner, indicating that we are to replicate those three variables N times, indexed by i in this case. This is a common convention, and will be used in this thesis.

2.1.2 Inference

Given a graphical model, whether directed or undirected, a common task is to infer the distribution (or a point estimate) of a subset of variables $\mathbf{Q} = \{Q_i\}$, conditioned on some (observed) evidence variables $\mathbf{E}$. For example, if we know the political leanings of some pundits, what is the distribution over political leanings of the rest of them? The general problem of *graphical model inference* is intractable [Cooper, 1990]. However, many approximate inference techniques have been developed over the years, from belief propagation [Pearl, 1988, Yedidia et al., 2003] to variational methods [Jordan et al., 1999] to Markov Chain Monte Carlo (MCMC) approaches [Robert and Casella, 2005]. In Chapter 5 of this thesis, we derive a Gibbs sampler (cf. [Geman and Geman, 1984]) to perform inference in our model, which is a special case of the latter. In this section, we briefly describe the main ideas behind Gibbs sampling, and direct the reader to one of many excellent surveys and books to get a more detailed picture (cf. [Casella and George, 1992, Fox, 2009, Geman and Geman, 1984, Gilks et al., 1995, Koller and Friedman, 2009, Robert and Casella, 2005]).

However, before describing Gibbs sampling, we must address a fundamental question: how does sampling help us with graphical model inference? To answer this, we consider a common inference scenario of computing the posterior mean of a function of some random variables $\mathbf{Q}$:

$$\widehat{\mathbf{Q}} = \int f(\mathbf{Q})\, dP_{\mathcal{G}}(\mathbf{Q}|\mathbf{E}), \tag{2.2}$$

where $P_{\mathcal{G}}$ is the distribution associated with the graphical model $\mathcal{G}$. While computing such an integral is, in general, intractable, it is often easier to sample from the distribution $P_{\mathcal{G}}$ directly. In this setting, if our samples from $P_{\mathcal{G}}$ are independent, then we know by the Strong Law of Large Numbers that the following sample mean converges almost surely to the true posterior expectation in Eq. (2.2):

$$\widehat{\mathbf{Q}} \approx \frac{1}{B} \sum_{b=1}^{B} f\left(\mathbf{Q}^{(b)}\right), \tag{2.3}$$

where $\mathbf{Q}^{(1)}, \mathbf{Q}^{(2)}, \ldots, \mathbf{Q}^{(B)}$ are B independent samples of the query variables.

We have now replaced the difficult integration problem with an easier sampling problem. The Gibbs Sampling algorithm [Geman and Geman, 1984] tells us how to sample from a potentially complicated distribution $P_{\mathcal{G}}$. In particular, in our setting, a Gibbs sampler produces a sequence of samples of the variables $\mathbf{Q}$, such that the sequence is a Markov chain with stationary distribution equal to $P_{\mathcal{G}}$. Rather than simultaneously sampling all the variables that make up the query set $\mathbf{Q}$, a Gibbs sampler will sequentially sample the conditional distribution of each variable Q_i given the rest. In a graphical model, these conditional distributions are often much easier to write down (and sample from) than the full joint over $\mathbf{Q}$, and thus Gibbs sampling is a popular method for graphical model inference.

Succinctly, the Gibbs Sampling algorithm is as follows:

Randomly initialize the variables to be sampled $\{Q_i^{(0)}\}$.
for samples $b = 1, \ldots, B$ **do**
  for each variable $Q_i \in \mathbf{Q}$ **do**
    sample $Q_i^{(b)}$ from $P(Q_i|\mathbf{Q}_{-i}^{(b-1)})$.

Many variations of this vanilla Gibbs sampler exist, e.g., handling cases when the conditional distributions are difficult to sample from (by incorporating Metropolis-Hastings steps [Hastings, 1970, Metropolis et al.,

1953]), sampling entire blocks of variables simultaneously when possible to improve mixing [Ishwaran and James, 2001], and so on.

2.2 Sparsity in Machine Learning

As discussed in Chapter 1, many domains today face a torrent of information, from computational biology to information retrieval. Consequently, in machine learning and statistics, we often find ourselves in this regime of big data, where problems have many more features than training examples (e.g., large vocabularies in text corpora, tens of thousands of voxels in a functional brain scan, etc.). One common approach for dealing with such challenges is to enforce or incentivize *sparsity* in the solution. For example, a topic from a topic model [Blei and Lafferty, 2009] may only be about a few words out of a large vocabulary, a linear predictor may only require a small number of nonzero weights out of a large feature set [Tibshirani, 1996], and an image might be represented by only a few visual features [Olshausen and Field, 1996].

Such sparsity can be desirable for many reasons. First and foremost, with sparsity we often get significant gains in efficiency, both computational and statistical. Given a fixed amount of data and fixed computational resources, if we only have to store, process and estimate a relatively small number of parameters, we can potentially do so more quickly and more accurately. Second, along with sparsity often comes interpretability. A parsimonious explanation of natural phenomena is more interpretable to domain experts, and as Occam's Razor would tell us, is more desirable philosophically and probabilistically.

In this section, we give a brief overview of the two primary frameworks for incentivizing sparsity in machine learning: (1) penalized loss minimization; and, (2) sparse Bayesian methods. We use both of these approaches later in this thesis (Chapter 6 and Chapter 5, respectively).

2.2.1 Penalized Loss Minimization

To motivate this approach, let us consider the classic problem of linear regression. Here, we assume we have a data set of N items $\{(\mathbf{x_i}, y_i)\}$, each made up of p input variables (or covariates), $x_{i1}, x_{i2}, \ldots, x_{ip}$, and a single, real output y_i. Our task is to learn a predictor that, given new input data, can accurately predict the output. It is straightforward to fit a standard linear model to this data by minimizing the squared residual:

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2, \tag{2.4}$$

where $\mathbf{y} = [y_1, y_2, \ldots, y_N]^\mathsf{T}$ is an $N \times 1$ vector, $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}]^\mathsf{T}$ is an $N \times p$ matrix, and $\boldsymbol{\beta}$ is a p-dimensional coefficient vector. However, if $p \gg N$, solving such an optimization will lead us to likely overfit to the training data. Moreover, a dense solution where all p values in $\boldsymbol{\beta}$ are nonzero can be hard to interpret.

One approach to address this concern is to directly enforce sparsity in our objective function. If we define the ℓ_0 pseudo-norm of a vector $\boldsymbol{\beta}$ as the number of nonzero elements in $\boldsymbol{\beta}$, then we can write our new objective function as follows:

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_0 \quad \text{where} \quad \mathbf{y} = \mathbf{X}\boldsymbol{\beta}. \tag{2.5}$$

While the direct control over sparsity expressed by Eq. (2.5) is desirable, this objective is intractable to solve, due to the combinatorial subset selection problem involved [Natarajan, 1995]. However, what if we consider a small modification to this objective, replacing the ℓ_0 pseudo-norm with the ℓ_1 norm?

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_1 \quad \text{where} \quad \mathbf{y} = \mathbf{X}\boldsymbol{\beta}. \tag{2.6}$$

The objective function in Eq. (2.6) is convex, incentivizes (rather than enforces) sparsity, and, perhaps surprisingly, under certain conditions, is *guaranteed* to give the same solution as Eq. (2.5) [Candès et al., 2006, Chen and Donoho, 1994, Chen et al., 2001, Stodden, 2006].

Commonly, a variation of Eq. (2.6) is instead written as a penalized optimization, as follows:

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1, \tag{2.7}$$

where λ is a regularization parameter, indicating how strong sparsity should be incentivized in the objective. In this setting of linear regression, this objective is known as the Lasso method [Tibshirani, 1996]. An identical ℓ_1 penalty term can be found in many other objective functions throughout machine learning, including logistic regression [Koh et al., 2007, Lee et al., 2006], sparse coding [Olshausen and Field, 1996] and dictionary learning [Mairal et al., 2010], all with the incentive of producing a sparse result vector $\boldsymbol{\beta}$.

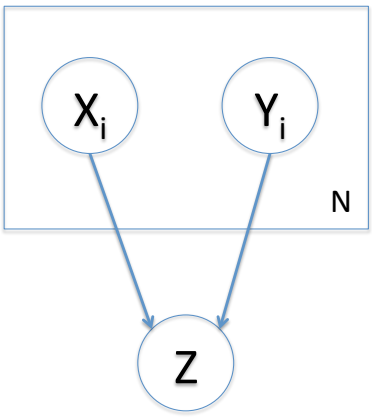
Structured Sparsity

In many real world settings, structure exists among the different input variables. For example, in a functional brain scan, such as an fMRI image, it is naïve to treat the voxels as all being independent of each other; neighboring voxels correspond to neighboring areas of the brain, and if we have a covariate per voxel, we might want to specify that voxels in the same functional brain region (e.g., visual cortex) are simultaneously either all active or all zero, but not a mixture of both.

Yuan and Lin introduced the Group Lasso—an elegant extension of the Lasso for precisely this setting—by making simple but critical modifications to Eq. (2.7) [Friedman et al., 2010, Yuan and Lin, 2007]. Here, we assume that the covariates $1, \ldots, p$ are partitioned into L groups, where group ℓ has p_ℓ covariates, and that we know these partitions *a priori*. Given this assumption, we can write the group lasso objective as follows:

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \sum_{\ell=1}^{L} \mathbf{X}_\ell\boldsymbol{\beta}_\ell\|_2^2 + \lambda \sum_{\ell=1}^{L} \sqrt{p_\ell}\|\boldsymbol{\beta}_\ell\|_2, , \tag{2.8}$$

where the subscript ℓ selects the covariates associated with group ℓ, and the $\sqrt{p_\ell}$ term is to normalize across potentially different group sizes. The sum of ℓ_2 norms acts like the Lasso penalty at the group level, incentivizing a small number of groups to be non-zero, but the covariates in the same group are either all



(a)

for all $i = 1, \ldots, N$ **do**
$X_i \sim \text{Normal}(0, 1)$
$Y_i \sim \text{Bernoulli}(0.1)$
$Z \sim \text{Normal}(\sum_{i=1}^{N} X_i \cdot Y_i, 1)$

(b)

Figure 2.3: A simple example of how sparsity can be directly encoded into a graphical model.

zero or all non-zero simultaneously. Jacob et al. extended this model to deal with overlapping groups [2009], while Friedman et al. extended it with an additional within-group sparsity term [2010].

Recent work in structured sparsity has led to methods that allow for richer sparsity structure to be specified in the optimization. For example, the approach we use in Chapter 6 of this thesis is the *graph-guided fused lasso* [Chen et al., 2012, Kim et al., 2009]. In this method, the structure of the problem is encoded as a graph $\mathcal{G}$ over the covariates, rather than simple groups, and the objective incentivizes covariates connected with a strongly weighted edge to have similar coefficients:

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_\beta \|\boldsymbol{\beta}\|_1 + \lambda_{\mathcal{G}} \sum_{(s,t) \in E(\mathcal{G})} w_{st} |\beta_s - \beta_t|, \tag{2.9}$$

where w_{st} is the strength of edge (s, t) in the edge set of $\mathcal{G}$. The graph fusion regularization encourages β_s to be close to β_t for all edges (s, t) in the graph where the strength of the regularization is proportional to the weight of the edge. In this way, highly related covariates, being closely connected in the graph by heavily weighted edges, would be incentivized to simultaneously have either all zero or all non-zero values in entries of $\boldsymbol{\beta}$. The graph fusion regularization parameter, $\lambda_{\mathcal{G}}$, regulates how big a role the graph should play in imposing structure on $\boldsymbol{\beta}$. This method is illustrated in more detail in Chapter 6.

2.2.2 Sparse Bayesian Methods

The sparse methods described so far in this section have been predominantly frequentist in nature. However, Bayesian statistics has its own rich set of tools for incentivizing and enforcing sparsity. We start with what is perhaps the most natural Bayesian equivalent to our subset selection problem from Eq. (2.5): the *spike and slab prior* [Ishwaran and Rao, 2005, Mitchell and Beauchamp, 1988]. In such a setting, we place a prior distribution on each β_j such that there is a discrete mass at $\beta_j = 0$. Generalizations of this approach lead to natural Bayesian extensions of the Lasso—Tibshirani himself indicated that the optimal solution to the Lasso problem can be viewed as a *maximum a posteriori* (MAP) inference in a Bayesian network with independent Laplace priors on the covariates β_j [Tibshirani, 1996]. These generalizations almost universally are based on the idea of scale mixtures of Normal distributions (cf. [Armagan et al., 2011, Carvalho et al., 2009, Griffin and Brown, 2010, Hans, 2009, MacLehose and Dunson, 2010, Park and Casella, 2008, Polson and Scott, 2010]).

Moving on from direct generalizations of the Lasso, we consider the simple idea of enforcing sparse structure in a probabilistic graphical model, which is what we do in Chapter 5 of this thesis. If we consider Figure 2.3, we see a simple model where the value of Z is based on a small fraction of the variables $\{Y_i\}$, as specified by the random mask $\{X_i\}$. This graphical model exhibits *context-specific independence*, because Z is independent of each variable Y_i where $X_i = 0$.[5]

[5]Such notions of sparsity are easily extended to the Bayesian nonparametric setting via the Beta/Bernoulli process [Griffiths and Ghahramani, 2005, Thibaux and Jordan, 2007].

Chapter 3

Interactive Concept Coverage with Simple Interactions

TIME Magazine once asked, "How many blogs does the world need?" [Kinsley, 2008], claiming that there were too many. Indeed, the blogosphere has experienced a substantial increase in the number of posts published daily.[1] One immediate consequence is that many readers now suffer from information overload.

While the vast majority of blogs are not worth reading for the average user, even the good ones are too many to keep up with. Moreover, there is often significant overlap in content among multiple blogs. To further complicate matters, many stories seem to resonate in the blogosphere to an extent that is largely uncorrelated with their true importance. For example, in the spring of 2007, Politico broke a story about John Edwards' $400 haircut in a blog post [Smith, 2007], which was almost instantly seized upon by the rest of the blogosphere. Over the next two weeks, the haircut story sparked several major online debates. Avoiding this story was difficult for most Web users, and nearly impossible for those interested in politics but not in this particular line of debate.

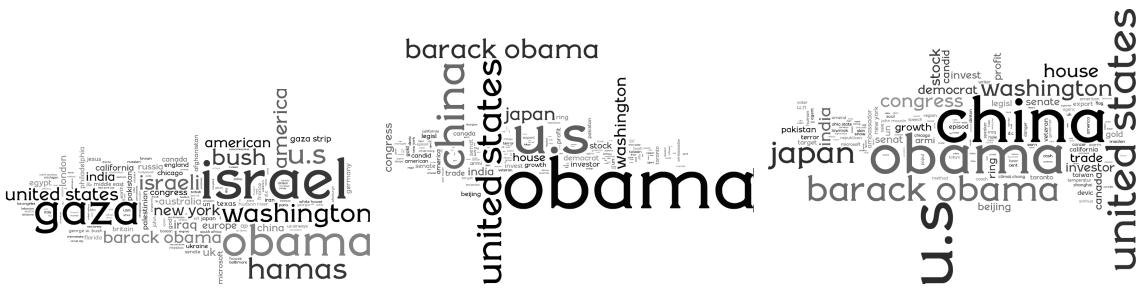
In this chapter, we demonstrate how we can use the interactive concept coverage methodology, as described in Section 1.2, to successfully address this problem. We will describe in detail how the steps of the methodology can be applied in this setting to effectively recommend a small set of relevant blog posts to a user, covering the important stories of the day and personalized to the user's individual tastes.[2] In addressing this problem of information overload in the blogosphere, we will be able to demonstrate how our methodology can work at its most basic level: in a *queryless* setting with *simple user interactions*.

First, however, we must ask: why is interactive concept coverage an appropriate approach for this task? Recommending blog posts has many of the hallmarks of the problems we described in the previous chapter that motivated our methodology:

- Reading news articles and blog posts is a naturally queryless endeavor, and thus keyword search is not an appropriate solution.
- People have different interests and tastes, and so showing everyone the same set of blog posts is not optimal.

[1] `http://en.wordpress.com/stats/posting/`

[2] We will defer discussing queries and the personalized affinity function (Steps 2 and 4 of our methodology) until Chapter 4.



(a) (b) (c)

Figure 3.1: (a) Global term frequency across the blogosphere (January 17, 2009). The size of a term is proportional to its frequency. (b) Coverage vs. (c) incremental coverage of a post about Obama and China, given that we already saw a post about Obama. The incremental coverage of Obama is much smaller than the regular coverage.

- According to the Pew Research Center's Project for Excellence in Journalism, nearly a quarter of all Americans consumed news on their mobile devices in 2011,[3] where screen real estate is at a premium. Moreover, this proportion is rising. The cost of displaying redundant news articles or blog posts to such users is high, and thus any recommendation system for news and blogs should incentivize diversity.
- It is natural for users to give simple like/dislike feedback on news articles and blog posts, and thus this recommendation task is an ideal initial testbed for our personalization methodology.

Having established that this news article and blog post recommendation domain is a reasonable fit for our approach, we can now instantiate the interactive concept coverage methodology and evaluate its performance on this task. In the remainder of this chapter, we will:

- Formally define potential *concept representations* for blogs and online news.
- Define a *document coverage function* that indicates how much a particular blog post or news article *covers* a concept. For example, an article about the 2012 American Presidential election might cover an "Obama" concept.
- Define a *set coverage function* that encodes an intuitive notion of diminishing returns, incentivizing us to pick a diverse set of relevant posts.
- Describe an optimization algorithm for *selecting a constrained set of posts efficiently and near-optimally*, with respect to a natural objective function.
- Describe an online learning algorithm for *learning a personalized concept weighting* from simple user feedback.

3.1 Concept Representation and Coverage

In order to reason about the textual content of a document collection, we need some manner of distilling each document into the concepts that it is about. For decades, the linguistics, machine learning and natural

[3] `http://stateofthemedia.org/2012/`

language processing (NLP) communities have proposed different representations of text, from low-level term-based representations such as the classic bag of words [Salton et al., 1975] model, to higher level topic models [Blei and Lafferty, 2009]. While in Chapter 6 we will introduce a novel concept representation of documents inspired by the challenges of personalization, in the balance of this thesis—including this chapter—we will incorporate simple variations of these common document representations into our framework to solve our problem of selecting relevant posts from the blogosphere. Specifically, we use a variety of simple "bag of features" models in our work, ranging from bags of words (Chapters 4) to bags of named entities and noun phrases (this chapter). It is important to note here that we do not explore the use of more advanced NLP techniques in this thesis, such as syntactic parsing, leaving this for future work. We do, however, occasionally use part-of-speech information to guide us with feature selection, and always simplify words into stems (e.g., "personalization" goes to "personal"), allowing us to consolidate related words to the same token.[4]

We now formally define a weighted concept representation as follows:

Definition 3.1.1 (Weighted Concept Representation)**.** *A* weighted concept representation *is a quadruple* $\langle \mathcal{C}, \mathcal{D}, cover_{\cdot}(\cdot), \mathbf{w} \rangle$. *$\mathcal{C}$ is a finite set of concepts and $\mathcal{D}$ is a finite set of documents. The relation between documents and concepts is captured by the* document coverage function*, $cover_j(i) : \mathcal{C} \rightarrow \mathbb{R}_{\geq 0}$, which quantifies the amount document $d_j \in \mathcal{D}$ covers concept $c_i \in \mathcal{C}$. $\mathbf{w} \in \mathbb{R}_{\geq 0}^{|\mathcal{C}|}$ is a weight vector over concepts, indicating the relative importance of each concept.*

In the simplest case, we can use terms in a vocabulary to represent our concepts $\mathcal{C}$, and we can define each concept weight w_i to reflect the prevalence of concept c_i in the document collection $\mathcal{D}$, thereby encoding the idea that covering, e.g., "Central Catholic High School" should not be as valuable as covering "Obama." For example, Figure 3.1(a) shows a typical day in the blogosphere (January 17, 2009), where the size of a term is proportional to its frequency across the blogosphere. Examining the picture, we can spot some of the popular stories for that day: the inauguration of Barack Obama and the Israel-Gaza conflict. If we define $cover_{\cdot}(\cdot)$ to be a binary indicator function, turning documents into subsets of concepts, we can imagine formalizing our post selection problem as the well-known *budgeted maximum coverage* problem [Khuller et al., 1999]:

Definition 3.1.2 (Budgeted Maximum Coverage)**.**
Given a set of ground elements $\mathcal{C}$, a weighting $\mathbf{w} \in \mathbb{R}_{\geq 0}^{|\mathcal{C}|}$, a collection $\mathbb{S} = \{S_1, ...S_m\}$ of subsets of $\mathcal{C}$, and a budget $k \geq 0$, select $\mathcal{A} \subseteq \mathbb{S}$ of size at most k which maximizes the total weight of the covered elements, $\bigcup\{S_j \in \mathcal{A}\}$.

In our setting, this coverage can be formalized as maximizing:

$$F(\mathcal{A}) = \sum_{i \in \mathcal{C}} w_i \mathbf{1}(\exists d_j \in \mathcal{A} : \ cover_j(i) = 1),$$

subject to a budget of $|\mathcal{A}| \leq k$.

Although max-coverage is an NP-hard problem, there are several efficient and effective approximation algorithms for this task. However, this naïve approach suffers from two serious drawbacks:

- *Concept importance in document*: The binary document coverage function does not characterize how relevant a document is to a particular concept, e.g., a post about Obama's inauguration speech covers

[4] A more detailed description of document representations can be found in one of many books on computational linguistics (e.g., [Smith, 2011].)

Obama just as much as a post that barely mentions him. As a side effect, this objective rewards "name-dropping" posts (posts that include many concepts, without being about any of them, such as a page with the tour dates of a band).

- *Incremental coverage*: This objective function results in a notion of coverage that is too strong, since after seeing one post that covers a certain concept, we will never gain anything from another post that covers the same concept. This does not correspond to our intuitive notion of coverage: each additional time we see a concept we should get an additional reward, which decreases with the number of occurrences. For example, suppose we show the user a post about Obama's inauguration. The second post we consider showing her is about the effect of Obama's presidency on China. Figure 3.1(b) shows the raw coverage of the second post, and "Obama" is the top-covered concept. However, if we take into account the fact that we have already covered the concept "Obama" to some extent by the first post, the coverage by the second post changes. Figure 3.1(c) shows the *incremental coverage* by the second post. As illustrated, the significance of this post towards "Obama" is diminished, and most of our reward would come from covering "China."

We address these two problems in turn.

First, each document should exhibit different degrees of coverage for the concepts it contains, which can be achieved by softening the notion of coverage, $\textit{cover}_j(i)$. One approach is to use a probabilistic definition of coverage, as defined by, e.g., a generative model of text. In such a setting, we assume we are given $P(c_i \mid d_j)$, the probability of a concept given a document. If, for example, our concepts are topics discovered by a topic model, such as latent Dirichlet allocation (LDA) [Blei et al., 2003], then this term is simply the probability that document j is about topic i. More generally, any generative model for the particular set of concepts can be used to define this probability.

Given such a probabilistic model, we can define the notion of soft coverage more formally. However, first, we must consider that concepts occur at different levels of granularity, and that, intuitively, a document is likely to cover more concepts in a fine-grained concept representation than in a coarse-grained concept representation. For example, a single document might cover only one or two topics from a topic model (e.g., topics representing the inauguration and the Middle East), whereas it might cover over a dozen words and phrases (e.g., "Obama," "White House," "politics," "Chief Justice," "Palestinians," "Israel," "Gaza," etc.). With this in mind, we can define our soft, probabilistic notion of coverage as:

$$\textit{cover}_j(i) = 1 - (1 - P(c_i \mid d_j))^{\ell}, \tag{3.1}$$

where ℓ is a parameter associated with the granularity of the concept representation. If we assume that, for each document d_j, we draw ℓ random concepts (with replacement) from its concept probability distribution, $P(c_i \mid d_j)$, then $\textit{cover}_j(i)$ is precisely defined to be the probability that concept c_i appears at least one time in this set.[5]

Now we move to the second problem described above, *Incremental coverage*.

Our probabilistic approach allows us to define the importance of concepts in individual posts. However, if we define coverage as $F(\mathcal{A}) = \sum_{i \in \mathcal{C}} w_i \sum_{d_j \in \mathcal{A}} \textit{cover}_j(i)$, then the *Incremental coverage* problem would persist, as this function does not possess the diminishing returns property. Instead, extending the probabilistic interpretation further, we can view set coverage as a sampling procedure: each document d_j

[5]It is important to note that, by defining $P(c_i \mid d_j)$ as a probability distribution that must sum to one over the entire space of concepts, we alleviate the problem of "name-dropping," since a post cannot cover a large number of features well.

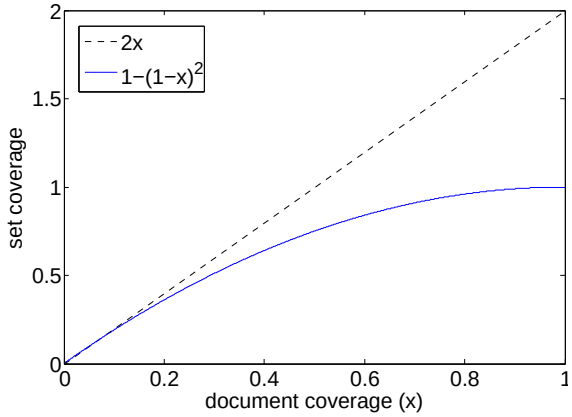


Figure 3.2: This plot shows how two documents who each identically cover a single feature with coverage probability x combine to jointly cover that feature, under both the probabilistic set coverage function defined in Eq. (3.2) and also using a simple additive combination. We note that, for small values of x, the two set coverage functions act linearly, with no diminishing returns.

tries to cover concept c_i with probability $\textit{cover}_j(i)$, and the concept is covered if at least one of the posts in $\mathcal{A}$ succeeds. Thus, as $\mathcal{A}$ grows, adding a post provides less and less additional coverage.

Formally, we can define the *probabilistic set coverage* of a concept by a set of documents $\mathcal{A}$ as:

$$\textit{cover}_{\mathcal{A}}(i) = 1 - \prod_{d_j \in \mathcal{A}} (1 - \textit{cover}_j(i)). \tag{3.2}$$

It is instructive to note here the importance of the concept granularity parameter, ℓ, for ensuring that our probabilistic set coverage function in fact gives us our desired diminishing returns property. Consider an example of two documents that cover a concept c_i by the same amount, $x \in [0, 1]$. For a given document coverage probability x, we can compute the set coverage of this concept by the two documents as $1 - (1 - x)^2$, which we have plotted in Figure 3.2 as a solid blue line. Note that for small values of x (i.e., small document coverage probabilities less than 0.2), we see that the behavior of the probabilistic set coverage function falls in a linear regime. This means that, for small values of x, we will not obtain diminishing returns, but rather observe additive behavior, as evident by comparing our coverage function to the additive coverage function plotted in black. As such, when we have many fine-grained concepts, such as words or named entities, that naturally have small concept probabilities, $P(c_i \mid d_j)$, the concept granularity parameter is needed to push the document coverage probability out of the linear regime and into the regime of diminishing returns.[6]

Finally, given our definition of set coverage, we propose the following objective function for the problem of probabilistic coverage of the blogosphere:

$$F(\mathcal{A}) = \sum_{i \in \mathcal{C}} w_i \textit{cover}_{\mathcal{A}}(i). \tag{3.3}$$

Our task is to find k posts maximizing the above objective function:

$$\mathcal{A}^* = \underset{\mathcal{A} \subseteq \mathcal{D}: |\mathcal{A}| \leq k}{\operatorname{argmax}} \; F(\mathcal{A}). \tag{3.4}$$

[6]Appendix 3.10 describes a simple heuristic based on this observation for defining ℓ. *Learning* this concept granularity parameter directly from user feedback is an interesting subject for future work.

3.2 Optimizing Set Coverage

Using the probabilistic notions of document coverage (Eq. (3.1)) and set coverage (Eq. (3.2)) in the previous section, our goal now is to find the set of posts $\mathcal{A}$ that maximizes our objective function in Eq. (3.3). Unfortunately, we can show by reduction from max-coverage that this objective is NP-complete, suggesting that the exact maximization of this function is intractable. However, our objective function satisfies an intuitive diminishing returns property, *submodularity*, which allows us to find good approximations very efficiently:

Definition 3.2.1 (Submodularity)**.** *A set function F is* submodular *if,* $\forall \mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}, \forall s \in \mathcal{V} \setminus \mathcal{B}$, $F(\mathcal{A} \cup \{s\}) - F(\mathcal{A}) \geq F(\mathcal{B} \cup \{s\}) - F(\mathcal{B})$.

Claim 3.2.1. *The probabilistic coverage objective function for the blogosphere in Eq. (3.3) is submodular.*

Proof. Let $\mathcal{B} \subseteq \mathcal{V}$ and $s \in \mathcal{V} \setminus \mathcal{B}$.

$$\begin{aligned}
cover_{\mathcal{B}\cup\{s\}}(i) - cover_{\mathcal{B}}(i) &= 1 - \prod_{j\in\mathcal{B}\cup\{s\}} (1 - cover_j(i)) - \left(1 - \prod_{j\in\mathcal{B}}(1 - cover_j(i))\right), \\
&= \prod_{j\in\mathcal{B}}(1 - cover_j(i)) - \prod_{j\in\mathcal{B}\cup\{s\}} (1 - cover_j(i)), \\
&= \prod_{j\in\mathcal{B}}(1 - cover_j(i)) \left(1 - (1 - cover_s(i))\right), \\
&= \prod_{j\in\mathcal{B}}(1 - cover_j(i)) \left(cover_s(i)\right).
\end{aligned}$$

Because $cover_j(i)$ is defined as a probability, it is in the range $[0, 1]$, and therefore $(1 - cover_j(i)) \in [0, 1]$ (for all j). Thus, for any $\mathcal{A} \subseteq \mathcal{B}$, we have that, $\prod_{j\in\mathcal{B}}(1 - cover_j(i)) \leq \prod_{j\in\mathcal{A}}(1 - cover_j(i))$. Hence,

$$\begin{aligned}
\prod_{j\in\mathcal{B}}(1 - cover_j(i)) \left(cover_s(i)\right) &\leq \prod_{j\in\mathcal{A}}(1 - cover_j(i)) \left(cover_s(i)\right), \\
&= \prod_{j\in\mathcal{A}}(1 - cover_j(i)) - \prod_{j\in\mathcal{A}\cup\{s\}} (1 - cover_j(i)), \\
&= cover_{\mathcal{A}\cup\{s\}}(i) - cover_{\mathcal{A}}(i).
\end{aligned}$$

Thus, $cover_{\mathcal{A}}(i)$ is submodular. Since submodularity is closed under nonnegative linear combinations, and our weights $w_i \geq 0$, it directly follows that our coverage function $F(\mathcal{A})$ is submodular. □

Intuitively, submodularity characterizes the notion that reading an article s after reading a small set of articles $\mathcal{A}$ provides more coverage than reading s after having already read the larger set $\mathcal{B} \supseteq \mathcal{A}$.

Although maximizing submodular functions is NP-hard [Khuller et al., 1999], by discovering this property in our problem, we can take advantage of several efficient approximation algorithms with theoretical guarantees. For example, the classic result of Nemhauser et al. [1978] shows that by simply applying a greedy algorithm to maximize our objective function in Eq. (3.3), we can obtain a $(1 - \frac{1}{e})$ approximation of the optimal value. Thus, a simple greedy optimization can provide us with a near-optimal solution. However, since the number of blog posts published daily is quite large, a naïve greedy approach can be

too costly. Therefore, we use the Cost-Effective Lazy Forward Selection (CELF) algorithm [Leskovec et al., 2007], which provides the same approximation guarantees, but uses lazy evaluations, often leading to dramatic speedups.[7]

3.3 Personalizing with Simple Interaction

To this point, we have described how we can use our interactive concept coverage methodology to address the problem of information overload in the blogosphere, formalizing the notions of a weighted concept representation, document coverage and set coverage. Thus far, we have assumed a fixed, global set of concept weights $\mathbf{w}$, implying a universal set of preferences over concepts. However, in reality, each user has different interests, and using a global definition of what is important could result in articles being selected and presented to a user containing many topics that do not interest him or her. For example, one user might care about a "NASCAR" concept, while others may be indifferent to it. Our goal in this section is to utilize simple user feedback—liking or disliking an article—to *learn* a personalized notion of concept importance for each user.

Recall that, in the previous section, $F(\mathcal{A})$ assigns a fixed weight w_i to every concept, representing its importance. Here, we will augment the fixed weights w_i with personalized preferences π_i for each feature i.

In the following, we assume that a user's coverage function is of the form:

$$F_{\boldsymbol{\pi}^*}(\mathcal{A}) = \sum_{i \in \mathcal{C}} \pi_i^* \; w_i \; \mathit{cover}_{\mathcal{A}}(i), \tag{3.5}$$

for some unknown set of weights $\{\pi_i^*\}$. Our goal now is to learn a user's coverage function $F_{\boldsymbol{\pi}^*}(\mathcal{A})$ by learning the optimal personalized preferences $\{\pi_i^*\}$ over the concepts $\mathcal{C}$.

3.3.1 Interaction Models

In order to receive personalized results, users need to communicate their preferences. Since $F_{\boldsymbol{\pi}}$ is a set function, the most natural notion of feedback from a machine learning perspective would be for users to provide a single label for the set of posts that they are presented, indicating whether they like or dislike the entire set. However, this approach suffers from two limitations. First, from the point of view of the user, it is not very natural to provide feedback on an entire set of posts. Second, since there are exponentially many such sets, we are likely to need an extensive amount of user feedback (in terms of sets of posts) before we could learn this function. Instead, we assume that users go through a list of posts $\mathcal{A}$ in order, submitting feedback f_j ("liked"= +1, "indifferent" = 0, "disliked" = -1) for each post $d_j \in \mathcal{A}$. We take no feedback on a post to mean "indifferent."

3.3.2 Personalization by Minimizing Regret

Our objective function is defined in terms of sets, but our feedback is in terms of individual posts. How should we provide an appropriate credit assignment?

[7] Chapter 5 of Krause's doctoral thesis provides an excellent overview of submodularity, including pseudocode of the CELF algorithm [2008].

One possible solution would be to assume that the feedback that a user provides for a particular post is independent of the other posts presented in the same set. In this case, one can view the user feedback as being labeled data on which we can train a classifier to determine which posts the user likes. However, this assumption does not fit with our interaction model, as a user might not like a post either because of its content or because previous posts have already covered the story.

To address this issue, we consider the *incremental coverage* of a post, i.e., the advantage it provides over the previous posts. The incremental coverage we receive by adding post d_j to the set $\mathcal{A}$ is:

$$\textit{inc-cover}_j(\mathcal{A}, i) = \textit{cover}_{\mathcal{A} \cup d_j}(i) - \textit{cover}_{\mathcal{A}}(i).$$

Note that if $\textit{cover}_{\mathcal{A}}(i)$ is defined as in Eq. (3.2), then the incremental coverage is the probability that d_j is the first post to cover feature c_i, given that we have already seen the posts in $\mathcal{A}$. Furthermore, if we view the set of documents $\mathcal{A}$ as an ordered set $\mathcal{A} = \{d_1, \ldots, d_k\}$,[8] the sum of incremental coverages is a telescoping sum that yields the coverage of a set of documents $\mathcal{A}$:

$$\begin{aligned}\sum_{d_j \in \mathcal{A}} \textit{inc-cover}_j(d_{1:j-1}, i) &= \sum_{d_j \in \mathcal{A}} \textit{cover}_{d_{1:j}}(i) - \textit{cover}_{d_{1:j-1}}(i), \\ &= \textit{cover}_{\mathcal{A}}(i),\end{aligned}$$

where $d_{1:j-1}$ is shorthand for the set of documents $\{d_1, \ldots, d_{j-1}\}$.

Using incremental coverages, we define an intuitive reward we receive after presenting $\mathcal{A}$ to a user with preferences $\boldsymbol{\pi}$ and obtaining feedback $\mathbf{f}$:

$$Rew(\boldsymbol{\pi}, \mathcal{A}, \mathbf{f}) = \sum_{i \in \mathcal{C}} \pi_i \, w_i \sum_{d_j \in \mathcal{A}} f_j \;\; \textit{inc-cover}_j(d_{1:j-1}, i).$$

If the user liked all of the documents in $\mathcal{A}$ (i.e., $\forall j, f_j = 1$), this reward becomes exactly the coverage function we are seeking to maximize, $F_{\boldsymbol{\pi}}(\mathcal{A}) = \sum_{i \in \mathcal{C}} \pi_i \, w_i \, \textit{cover}_{\mathcal{A}}(i)$, as in Eq. (3.5).

Our algorithm maintains an estimate of the user's preferences at each time step t, $\boldsymbol{\pi}^{(t)}$. Given this estimate, we optimize $F_{\boldsymbol{\pi}^{(t)}}(\mathcal{A})$ and pick a set of documents $\mathcal{A}^{(t)}$ to show the user. After receiving feedback $\mathbf{f}^{(t)}$, we gain a reward of $Rew(\boldsymbol{\pi}^{(t)}, \mathcal{A}^{(t)}, \mathbf{f}^{(t)})$. After T time steps, our average reward is therefore:

$$AvgRew(T) = \frac{1}{T} \sum_{t=1}^{T} Rew(\boldsymbol{\pi}^{(t)}, \mathcal{A}^{(t)}, \mathbf{f}^{(t)}).$$

Since our decisions at time t can only take into account the feedback we have received up to time $t-1$, the decisions we made may have been suboptimal. For comparison, consider the reward we would have received if we had made an informed choice for the user's preferences $\boldsymbol{\pi}$ considering all of the feedback from the T time steps:

$$BestAvgRew(T) = \max_{\boldsymbol{\pi}} \frac{1}{T} \sum_{t=1}^{T} Rew(\boldsymbol{\pi}, \mathcal{A}^{(t)}, \mathbf{f}^{(t)}). \tag{3.6}$$

That is, after seeing all the user feedback, what would have been the right choice for user preference weights $\boldsymbol{\pi}$? The difference between our reward and this best choice in retrospect is called the *regret*:

[8] This ordering could be defined by the order the posts are presented to the user, e.g., the one picked by the greedy algorithm.

Definition 3.3.1 (Regret)**.** *Our average regret after T time steps is the difference $BestAvgRew(T) - AvgRew(T)$.*

Positive regret means that we would have preferred to use the weights $\boldsymbol{\pi}$ that maximize Eq. (3.6) instead of our actual choice of weights $\boldsymbol{\pi}^{(t)}$. A *no-regret learning algorithm*, such as the one we describe below, will allow us to learn $\boldsymbol{\pi}^{(t)}$ such that, as T goes to infinity, the regret will go to zero at a rapid rate. Intuitively, this no-regret guarantee means that we learn a sequence $\boldsymbol{\pi}^{(t)}$ that does as well as any fixed $\boldsymbol{\pi}$—including the true user preferences, $\boldsymbol{\pi}^*$—on the sets of posts that the user is presented. By learning the personalized coverage function for a particular user in this manner, the posts we provide will be tailored to his tastes.

3.3.3 Learning a User's Preferences

We now describe our algorithm for learning $\boldsymbol{\pi}^*$ from repeated user feedback sessions. Like many online algorithms [Cesa-Bianchi and Lugosi, 2006], our approach updates our estimated $\boldsymbol{\pi}^{(t)}$ using a multiplicative update rule. In particular, our approach can be viewed as a special case of Freund and Schapire's multiplicative weights algorithm [Freund and Schapire, 1999].

The algorithm starts by choosing an initial set of weights $\pi^{(1)}$. (Without loss of generality, we assume weights are normalized to sum to 1, since the coverage function is insensitive to scaling.) In the absence of prior knowledge about the user, we can choose the uniform distribution:

$$\pi_i^{(1)} = \frac{1}{|\mathcal{C}|}.$$

If we have prior knowledge about the user, we can start from the corresponding set of weights.

At every round t, we use our current distribution $\boldsymbol{\pi}^{(t)}$ to pick k posts, $\mathcal{A}^{(t)}$, to show the user. After receiving feedback $\mathbf{f}^{(t)}$, we would like to increase the weight of features covered by posts the user liked, and decrease the weight of features covered by posts the user disliked. These updates can be achieved by a simple multiplicative update rule:

$$\pi_i^{(t+1)} = \frac{1}{Z}\,\pi_i^{(t)}\,\left(\frac{1}{\beta}\right)^{\mathcal{M}(i,\mathbf{f}^{(t)})}, \tag{3.7}$$

where Z is the normalization constant, $\beta \in (0,1)$ is the inverse learning rate, and, intuitively, $\mathcal{M}(i,\mathbf{f}^{(t)})$ measures the contribution (positive or negative) that feature i had on our reward:

$$\mathcal{M}(i,\mathbf{f}^{(t)}) := \frac{w_i \sum_{d_j \in \mathcal{A}^{(t)}} f_j^{(t)}\ \textit{inc-cover}_j(d_{1:j-1}, i)}{2\max_i w_i}, \tag{3.8}$$

where the normalization by $2\max_i w_i$ is simply used to keep this term in the range $[-0.5, 0.5]$.

If the learning rate $1/\beta$ is large, we make large moves based on the user feedback. As the learning rate tends to 0, these updates become less significant. Thus, intuitively, we will start with a small value of β and slowly increase it.

Claim 3.3.1. *If, for number of personalization epochs T, we use an inverse learning rate β_T given by:*

$$\beta_T := \frac{1}{1+\sqrt{\frac{2\ln|\mathcal{C}|}{T}}}, \tag{3.9}$$

then our preference learning procedure will have regret bounded by:

$$BestAvgRew(T) - AvgRew(T) \leq \mathcal{O}\left(\sqrt{\frac{\ln |\mathcal{C}|}{T}}\right).$$

Since our regret goes to zero as T goes to infinity, our approach is called a no-regret algorithm. The proof follows from Freund and Schapire [1999], by formalizing our learning process as a two-player repeated matrix game involving our algorithm and the user. (More details can be found in Appendix 3.8.)

3.4 Experimental Results

In this section, we evaluate the effectiveness of our methodology at **T**urning **D**own the **N**oise (TDN) in the blogosphere, by analyzing real blog data collected over a two week period in January 2009. These posts come from a diverse set of blogs, including personal blogs, blogs from mainstream news sites, commercial blogs, and many others.

We obtain the data from Spinn3r,[9] which at the time of these experiments, indexed and crawled 12 million blogs, collecting approximately 500,000 posts per day. After performing some simple data cleaning steps, such as removing web forums and classifieds, we reduce this number to about 200,000 posts per day in our data set. However, as this is real Web data, it is still invariably noisy even after cleaning. Thus, our algorithm must be robust to content extraction problems.

For each post, we extract named entities and noun phrases using the Stanford Named Entity Recognizer [Finkel et al., 2005] and the LBJ Part of Speech Tagger [Rizzolo and Roth, 2007], respectively. We remove infrequent named entities and uninformative noun phrases (e.g., common nouns such as "year"), leaving us with a total vocabulary size of nearly 3,000. (More details can be found in Appendix 3.9.)

We evaluate two instantiations of our model, one with a high-level, coarse-grained concept representation and one with a low-level, fine-grained concept representation:

1. In our high-level concept representation, our concepts are topics from an LDA topic model [Blei et al., 2003], learned on the noun phrases and named entities described above. We take the global importance weight of each topic, w_i, to be the fraction of terms in the corpus assigned to that topic. As this is a coarse-grained representation, we use a concept granularity of $\ell = 1$, thus directly defining $\textit{cover}_j(i) = P(c_i \mid d_j)$, which in the setting of topic models is the probability that d_j is about topic i. We use a Gibbs sampling implementation of LDA [Griffiths and Steyvers, 2004] with 100 topics and the default parameter settings. We refer to this instantiation as **TDN+LDA**.

2. For our low-level concept representations, we use the named entities and noun phrases directly. As this variant uses a lower-level concept set, we use a higher value for the concept granularity parameter, setting $\ell = 16$, which is approximately the mean number of occurrences of named entities and nouns per document in our corpus. We define $P(c_i \mid d_j)$ to be the probability of the term i given document j under the LDA model (i.e., after marginalizing out the topics). We refer to this instantiation as **TDN+NE**.

[9] `http://www.spinn3r.com`

3.4.1 Evaluating Coverage

The motivation as laid out in the beginning of this chapter was to select a set of blog posts that best covers the important and prevalent stories currently being discussed in the blogosphere. The major world events that took place during the time corresponding to our data set included the Israel-Gaza conflict, the inauguration of Barack Obama, the gas dispute between Russia and Ukraine, as well as the global financial crisis. As an example, here is the set of posts that our algorithm selects for an eight hour period on January 18, if our budget k is set to five:

1. Israel unilaterally halts fire as rockets persist
2. Downed jet lifted from ice-laden Hudson River
3. Israeli-trained Gaza doctor loses three daughters and niece to IDF tank shell
4. EU wary as Russia and Ukraine reach gas deal
5. Obama's first day as president: prayers, war council, economists, White House reception

The selected five posts all cover important stories from this particular day. The Israel-Gaza conflict appears twice in this set, due to its extensive presence in the blogosphere at the time. It is important to note, however, that these two posts present different aspects of the conflict, each being a prevalent story in its own right. By expanding the budget to fifteen posts, the algorithm makes additional selections related to other major stories of the day (e.g., George W. Bush's legacy), but also selects "lifestyle" posts on religion and cooking, since these represent the large portion of the blogosphere that is not directly related to news and current events.

As another example, here are the top five selected posts from the morning of January 23, the day after the Academy Award nominations were announced:

1. Button is top Oscar nominee
2. Israel rules out opening Gaza border if Hamas gains
3. Paterson chooses Gillibrand for U.S. Senate
4. Fearless Kitchen: Recipe: Medieval Lamb Wrap
5. How Obama avoided a misguided policy blunder

A post describing the Oscar-nominated movie *The Curious Case of Benjamin Button* supplants the Israel-Gaza conflict at the top of the list, while a cooking post makes it up to the fourth position.

We wish to quantitatively evaluate how well a particular post selection technique achieves the notion of coverage we describe above on *real blog data*. However, the standard information retrieval metrics of precision and recall are not directly applicable in our case, since we do not have labels identifying all the prevalent stories in the blogosphere on a given day and assigning them to specific posts. Rather, we measure the *topicality* of individual posts as well as the *redundancy* of a set of posts. We say a post is *topical* with respect to a given time period if its content is related to a major news event from that period. A post r is *redundant* with respect to a previous post p if it contains little or no additional information to

Figure 3.3: Topic representing the peanut butter recall from January 18, 2009, with the size of a term proportional to its importance in the topic.

post p. An ideal set of posts that covers the major stories discussed in the blogosphere would have high topicality and low redundancy.

We conducted a study on 27 users to obtain labels for topicality and redundancy on our data. We compared TDN+LDA and TDN+NE to four popular blog aggregation sites: the front page of Digg, Google Blog Search,[10] Nielsen BuzzMetrics' BlogPulse, and Yahoo! Buzz. Additionally, we also examine the performance of simpler objective functions on the post selection task.

Measuring Topicality

In order for users to measure the topicality of a blog post, they need an idea of what the major news stories are from the same time period. We express this information to our study participants by providing them with headlines gathered from major news sources in five different categories: world news, politics, business, sports, and entertainment. The headlines for each category are aggregated from three different news sources to provide a wider selection for the users and to avoid naming a single source as the definitive news outlet for a category. For instance, for politics we present headlines from Reuters, *USA Today*, and *The Washington Post*. This collection of headlines is akin to a condensed newspaper, and we refer to these stories as *reference stories*.

We present the participants with reference stories gathered at a particular time, e.g., January 18, 2009, 2:00pm EST, which we call the *reference time*. We then show each participant a set of ten posts that was chosen by one of the six post selection techniques, and ask them to mark whether each post is "related" to the reference stories. Each post is presented as a title along with a short description. The users are not made aware of which technique the posts come from, so as not to bias their ratings. The posts selected by TDN+LDA and TDN+NE were chosen from an eight hour window of data ending at the reference time, while the posts selected by the popular blog aggregation sites were retrieved from these sites within fifteen minutes of the reference time.

Figure 3.4(left) shows the results of the topicality user ratings on the six techniques. On average, the sets of ten posts selected by Google Blog Search, TDN+LDA and Yahoo! Buzz each contain five topical posts out of ten presented. The topicality of these techniques is significantly better than that of TDN+NE, Digg and BlogPulse. BlogPulse selects the most linked-to posts of the day, which does not seem to be a good heuristic for covering the important stories. Many of these posts are technology how-to pages, such as "Help With Social Bookmarking Sites," the highest ranked post from January 18. Digg selects its top posts

[10]In early 2009, when this study was conducted, the front page of Google Blog Search (`http://www.google.com/blogsearch`) had a listing of top blog posts of the day.

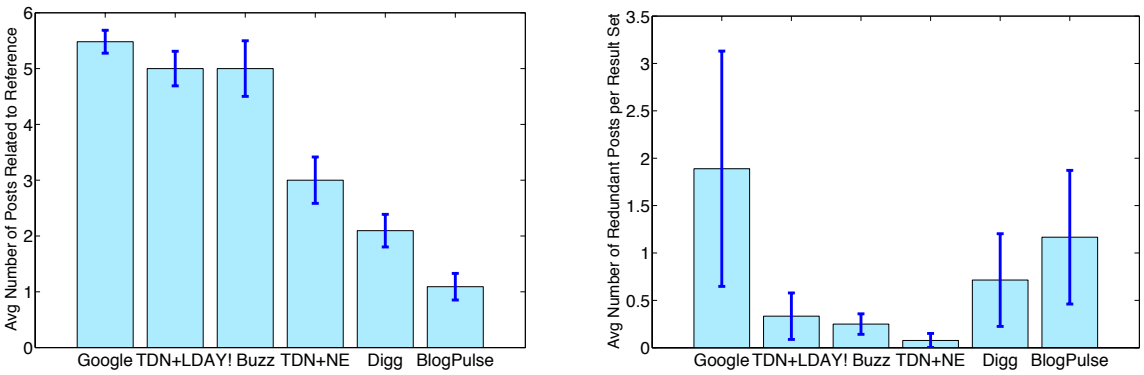


Figure 3.4: Left: Results from user study measuring topicality. The bars show the average number of posts (out of 10) that users found to be topical with respect to the reference stories. Right: Results of the redundancy user study. Users report the number of redundant posts for each post selection technique they are presented with. Error bars on all plots indicate standard error.

by user voting, and thus the top selected posts consist of a few prevalent stories and many entertaining or shocking posts, such as "Teen Stabbed But Makes It To Job Interview," the top post from February 6.

TDN+LDA outperforms TDN+NE because high-level concepts, such as LDA topics, capture stories in a better way than low-level concepts do. For example, for one eight hour period in our data set, there is a coherent LDA topic about the EU-Russia gas crisis. Therefore, when we cover this topic, we will present a story that is about the crisis. However, the named entity "Russia" may be covered by multiple stories. TDN+NE selects a post about Russia's plan to go ahead with the opening of a pediatric medical center in Moscow despite the current financial crisis, since it contains important named entities and nouns like "Russia," "Putin," "crisis," etc. Hence, if we only cover low-level concepts, we might select a post that is not topical, yet contains multiple important concepts.

While topicality captures a major aspect of our notion of coverage, in that important current events are covered by the selected posts, one drawback of this evaluation method is that lifestyle blog posts are not adequately represented. It is difficult to define a set of reference sites that summarize the day's most important recipes or most prevalent do-it-yourself tips, for instance. Furthermore, in our case, we did not want to show our study participants more than five categories of reference stories, so as not to overwhelm them. As a result, a post related to an important technology story would likely not be considered topical, as we left this category out.

Measuring Redundancy

The user study described in the previous section allowed us to measure whether posts were topical or not. However, topicality is not enough to judge the goodness of a set of posts, since they may all be about the same story, and hence not interesting. Instead, we want the posts to be diverse, so that they capture all of the important stories in the blogosphere, as well as appeal to everyone's interests. As part of our user study, we asked users to look at a set of fifteen posts selected by one of the six previously described post selection techniques, and mark any occurrences they thought were redundant. Each of 27 participants was presented

with either two or three sets of posts generated by different algorithms over the same time period. The users were not aware of the sources of the posts.

Figure 3.4(right) shows that both variants of our algorithm outperform Digg, BlogPulse and Google Blog Search on the redundancy metric. In other words, our algorithm selects diverse sets of posts. This diversity is primarily due to the diminishing returns property of our objective function. If we have covered the important concepts of a story once, covering it again yields only a small reward. Google Blog Search has the highest number of redundant results, and has high variance, suggesting that on some days many of the posts on its front page are similar. In fact, on average, the posts selected by Google Blog Search are nearly six times as redundant as those selected by TDN+LDA.

However, it should be noted that performing well on the redundancy metric alone is not sufficient. For example, it may turn out that all the posts picked by an algorithm are non-redundant, but meaningless, and hence of no interest to a user. Thus, an algorithm needs to perform well on both the topicality and the redundancy metric in order for it to be useful.

TDN+LDA and Yahoo! Buzz were the two techniques that performed well in both metrics. However, while Yahoo! Buzz uses Web search trends, user voting and other features to select its posts, TDN+LDA achieves the same topicality and redundancy performance by selecting posts only using simple text features. Furthermore, TDN+LDA adapts its results to user preferences, as described in Section 3.4.2.

Alternative Objective Functions

As an alternative to the submodular objective function defined in Eq. (3.3), we consider two simpler objective functions.

LDA-based Modular Function. A *modular function* is an additive set function where each element is associated with a fixed score, and the value for a set $\mathcal{A}$ is the sum of the scores of the elements of $\mathcal{A}$. Since the score of a post does not depend on the other elements in the set, there is no incentive to select a diverse set of posts. The naïve way of selecting posts using LDA fits under this modular framework. We first pick the top k topics based on their weight in the corpus. For each one, we pick the post that covers it the most. In addition to the potential for redundancy mentioned above, this technique suffers from the fact that it commits to a topic irrespective of the quality of the posts covering it. Furthermore, even if a post covers multiple topics well, it might not be selected as there may be some posts that better cover each individual topic. Using a strictly submodular objective function alleviates these problems.

For example, if we define our concepts based on a 50-topic LDA model trained on an eight hour data set from January 18, the topic with the lowest weight is about the peanut butter recall, a major news story at this time (*cf.* Figure 3.3). Thus, if we select fifteen posts following the naïve LDA approach, we do not pick a post from this topic. However, the weight of this topic (0.019) is not much lower than the mean topic weight (0.020). Moreover, since this topic closely corresponds to a prevalent news story, many posts cover it with high probability. TDN selects such a post because, unlike the naïve LDA approach, it simultaneously considers both the topic weights and the post coverage probabilities.

Budgeted Maximum Coverage. Another simple objective function we consider is budgeted maximum coverage, introduced in Definition 3.1.2, but with each concept (in this case, noun phrases and named entities) weighted by its corpus frequency. Optimizing this objective leads to the aforementioned "name-dropping" posts. For example, on an eight hour data set from January 20, the second post selected

announces the schedule of a rock band's upcoming world tour, and thus completely covers the concepts, "Washington," "Boston," "New York," "London," "Rome," and a few dozen more cities and countries. Once this post has been selected, there is no further incentive to cover these concepts.

3.4.2 Personalization

There are two methods by which we evaluate how well our algorithm personalizes the posts it selects in response to user feedback. In one setting, we conduct a user study to directly measure how many of the presented posts a study participant would like to read. In the second setting, we simulate user preferences on a targeted set of blog posts and observe how our objective function $F(\mathcal{A})$ changes with respect to the unpersonalized case.

Preferences of Real Users

We divide our blog data into 33 eight hour segments (epochs), and pick a starting segment at random for a particular user. We present our user with a set of ten posts from his starting segment, selected using TDN+LDA. The posts are displayed as a title and short summary. The user is instructed to read down the list of posts and, one by one, mark each post as "would like to read," "would not like to read," or "indifferent." The user is told to make each decision with respect to the previous posts displayed in that set, so as to capture the notion of incremental coverage. For example, a user might be excited to read a post about Obama's inauguration appearing at the top slot in a particular result set, and thus would mark it as "like to read." However, if four other very similar posts appear below it, by the time he gets to rating the fifth inauguration post in a row, he will likely label it as "not like to read."

After each set of ten posts, our personalization algorithm uses the user ratings to update the weights $\pi^{(t)}$, and selects a personalized set of posts for the next epoch.[11] We also ask the user to mark his preferences on unpersonalized posts presented for the same epochs. The order in which these two conditions are presented is randomized. We repeat this process for a total of five epochs. As this is not a longitudinal study, and we do not wish it to be overly tedious for our participants, we accelerate the personalization process by using an inverse learning rate β of 0.5, corresponding to a short-term learning horizon (i.e., $T \approx 9$ from Eq. (3.9)).

Figure 3.5(a) shows the result of this study on twenty users. The vertical axis of the plot shows the average number of posts liked by a user in a single epoch. As one would expect, at epoch 0, when the posts are always unpersonalized, the number of liked posts is approximately the same between the personalized and unpersonalized runs. However, in just two epochs, the users already show a preference towards the personalized results.

If a user only prefers sports posts, personalization is easy, as the user's interests are narrow. In our study, however, the participants were simply instructed to rate posts with their own personal preferences. As people are often eclectic and have varied interests, this task is harder, but more realistic. Thus, it is notable that we are still able to successfully adjust to user tastes in very few epochs, showing a significant improvement over the unpersonalized case.

[11] For this study, we train a separate LDA model on each epoch. As topics tend to change from one epoch to the next, we employ a simple bipartite matching algorithm to map personalization weights across epochs. Chapter 6 discusses more advanced concept representations that can elegantly handle this dynamism over time.

If instead of asking users to rate posts according to their personal tastes, we ask them to pretend that they only want to read posts on a specific subject (e.g., India), we observe interesting qualitative behavior. Initially, the top posts selected are about the main stories of the day, including the Israel-Gaza conflict and the Obama inauguration. After a few epochs of marking any India-related posts as "like" and all others as "dislike," the makeup of the selected posts changes to include more posts about the Indian subcontinent (e.g., "Pakistan flaunts its all-weather ties with China"). This is particularly notable given that these posts appear relatively infrequently in our data set, and thus without personalization, are rarely selected. Also, while after enough epochs, stories about India eventually supplant the other major news stories at the top of the result set, the Israel-Gaza stories do not disappear from the list, due to their high prevalence. We believe this is precisely the behavior one would want from such a personalization setting.

Simulating Preferences

We consider the case of a hypothetical sports fan, who always loves to read any sports-related post. In particular, every day, he is presented with a set of posts from the popular sports blog FanHouse.com, and he marks that he likes all of them. We simulate such a user in order to empirically examine the effect of personalization on the objective function.

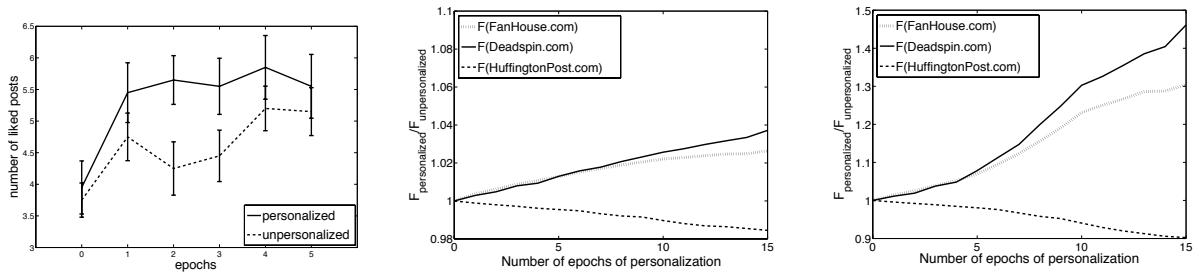
Specifically, we simulate this sports fan by marking all FanHouse.com posts as "liked" over a specified number of personalization epochs, updating the personalization weights $\pi^{(t)}$at each epoch. On the next epoch, which we call the evaluation epoch, we compute our objective function $F(\mathcal{A})$ on three different sets of posts. First, we compute $F(\mathcal{A})$ on the FanHouse.com posts from this epoch, hypothesizing that the more epochs we spend personalizing prior to the evaluation epoch, the higher this value will be. Second, we compute $F(\mathcal{A})$ on all the posts from DeadSpin.com, another popular sports blog. We also expect to see a higher value of our objective in this case. Finally, we compute $F(\mathcal{A})$ on all the posts from the HuffingtonPost.com Blog, a popular politics blog. The expectation is that by personalizing on sports posts for several days, $F(\mathcal{A})$ for a set $\mathcal{A}$ of politics posts will decrease with respect to the unpersonalized case.

Figure 3.5(b) shows the results of this experiment with a β value of 0.5, and we observe precisely the hypothesized behavior. The vertical axis of this plot shows the ratio of $F(\mathcal{A})$ computed with the learned personalization weights to that of $F(\mathcal{A})$ with the unpersonalized uniform weights, allowing us to compare across the three blogs. Thus, points on the plot that appear higher along the vertical axis than 1 indicate an improvement over the unpersonalized case, while any value below 1 indicates a decline with respect to the unpersonalized case.

Figure 3.5(c) shows the same simulation but with $\beta = 0.1$. This is an aggressive setting of the learning rate, and thus, as expected, the plot shows the objective function changing in the same direction but more rapidly when compared to Figure 3.5(b). These figures capture an important trade off for a deployed system, in that by varying the inverse learning rate β, we trade off the speed of personalization with the variety of selected posts.

3.5 Extensions

In this chapter, we give a no-regret guarantee on our algorithm for learning personalized concept weights from user feedback. However, a stronger guarantee would be to show that the weights $\boldsymbol{\pi}^{(t)}$ not only do



(a) (b) (c)

Figure 3.5: (a) Results of the personalization user study, measuring how many posts each user liked, out of the ten presented by TDN+LDA in each epoch. The personalized line corresponds to an inverse learning rate $\beta = 0.5$. (b,c) Effect of number epochs spent personalizing to the simulated preferences of a sports fan on the objective function F, with respect to no personalization. F is evaluated on two sports blogs and one politics blog. Inverse learning rate $\beta = 0.5$ (b), 0.1 (c)

well on the sets of posts from which they were learned, but also on the posts that would have been selected had we used the true $\boldsymbol{\pi}^*$ as the user preference weights for each day. For example, consider a user who is interested in politics and sports, but is also passionate about bagpiping. We may never show him any bagpiping posts, since they are not likely to be common. Thus, we may never receive feedback that would allow us to accurately model this portion of the user's true preferences. Yue and Guestrin developed an extension to our model to specifically deal with this *exploration/exploitation tradeoff*, using a contextual bandit framework [2011]. In particular, this is a tradeoff between *exploring* more of the space of possible user preferences—perhaps at the risk of recommending suboptimal results in the short-term—and *exploiting* the current preference model we already have learned.

Furthermore, in this work, we assume an explicit feedback model, where a user is observed providing "like," "dislike," or "indifferent" feedback on the individual articles that he or she is presented with. While we can apply this setup to implicit feedback data (e.g., all clicks count as "like," everything else is a "dislike"), it conflates indifference towards an article with disliking it. Recent work by Ahmed et al. extends our model to address this concern, using a more complex model of user viewing behavior [2012b].

3.6 Related Work

There are two categories of related work to consider in this chapter. First, we will look at alternative methods for selecting a diverse set of documents from a corpus. Second, we will look at various alternative methods for combating information overload in the blogosphere.

3.6.1 Incentivizing Diversity

When tasked with selecting a diverse set of documents from a large corpus, perhaps the most natural idea is to first *cluster* the posts, where posts in the same cluster cover the same concepts. Then, given clusters, we can pick a representative post from each of the k largest clusters. Such clustering approaches are common in the literature [Zhang et al., 2005]. However, most clustering methods require us to compute the distance

between every pair of posts, which amounts to $O(n^2)$ comparisons for n posts. Due to the sizable amount of posts published daily, methods that require $O(n^2)$ computation are practically infeasible. Our first desirable property for a coverage function is *scalability*, i.e., we should be able to evaluate coverage in time linear in the number of posts.

Early work by Carbonell and Goldstein [1998] introduced the concepts of "relevant novelty" and "marginal relevance" for retrieving diverse result sets in response to a user query. A line of related research is the area of subtopic retrieval [Chen and Karger, 2006, Zhai et al., 2003], where the task is to retrieve documents that cover many subtopics of the given query. For example, in response to the query "apple," such an algorithm would return documents covering the different subtopics that might go along with this query, including both the consumer electronics company and the fruit. In the traditional information retrieval setting, it is assumed that the relevance of each document is independent of the other documents. However, in subtopic retrieval the utility of a document is contingent on the other retrieved documents. In particular, a newly retrieved document is relevant only if it covers subtopics other than the ones covered by previous documents. Thus, the concept of relevance in subtopic retrieval is similar to our notion of coverage, which has a diminishing returns characteristic. However, while subtopic retrieval is query-based, we intend to apply our framework to more general retrieval tasks, even those with no explicit query.

More recent work has investigated the use of structural support vector machines (SVMs) to predict diverse subsets of documents by directly training for subtopic diversity [Yue and Joachims, 2008]. Such an approach requires labeled training data indicating subtopic coverage, which may be difficult or costly to obtain in practice. More closely related to our approach is the line of recent research on using submodular coverage functions to solve various sensor selection and placement problems, including which blogs a reader should follow to keep up to speed with breaking news [Krause and Guestrin, 2009, Krause et al., 2008, Leskovec et al., 2007].

Finally, there is a popular line of current research in the machine learning community on using *determinantal point processes* (DPPs) and derivative models as Bayesian priors that incentivize diversity [Affandi et al., 2012, Gillenwater et al., 2012a,b, Kulesza, 2012, Kulesza and Taskar, 2010, 2011a,b]. DPPs concisely capture the notion of *probabilistic mutual exclusion*, where similar objects (as determined by a kernel) are unlikely to appear together. In particular, the k-DPP model is quite related to our approach, as it defines a probability distribution over diverse sets of size k [Kulesza and Taskar, 2011a]. If an appropriate kernel function is defined over concepts, then this work can be used as an alternative to our submodular coverage approach for selecting diverse documents.

3.6.2 Taming Information Overload in the Blogosphere

Recently, there has been an increase in the number of websites that index blogs and display a list of the most popular stories. Some examples of such websites are Google Blog Search, Yahoo! Buzz, Digg, Technorati, Reddit and Blogpulse. Some of these websites display posts without any manual intervention, e.g., Google Blog Search and Blogpulse. However, most of these websites display posts which have either been handpicked by editors or have been voted for by users of the website. Most websites that pick posts automatically use a combination of features such as link structure (Blogscope), trends in search engine queries (Yahoo! Buzz), and the number of times a post is emailed or shared. Currently, we are only using features derived from the text of the posts, although in the future we hope to incorporate the link structure between posts into our algorithm. Another key difference is that most of these websites lack the personalization functionality we provide.

Agarwal et al. address a problem similar to ours [2009]. Their task is to select four out of a set of sixteen stories to be displayed on the Yahoo! homepage. The sixteen stories are manually picked by human editors; hence, all are of high quality. The authors use click-through rate to learn online models for each article. Their setting differs significantly from ours, since we tackle the problem of selecting ten out of roughly 60,000 posts for each eight hour segment. Moreover, as described in Section 3.4, our data is very noisy, and we do not have access to click-through rates.

Leskovec et al. propose a solution to the problem of selecting which blogs to read in order to come across all the important stories quickly [Leskovec et al., 2007]. Although related to our problem, a fundamental difference is that instead of trying to select which blogs to read, we present the user with a selection of posts from various blogs. Moreover our approach is completely content based, whereas the approach of Leskovec et al. is based only on the links between blogs. In addition, we also incorporate personalization into our algorithm, which they do not.

There has also been extensive work on building models and analyzing the structure of the blogosphere. For example, Finin et al. [2008] present a model of information flow in the blogosphere. Blogscope is intended to be an analysis and visualization tool for the blogosphere. Unlike us, they are not trying to cover the blogosphere. Instead, Blogscope presents the user with a search interface, and suggests some related words based on the search query. They give a preference to words whose frequency increases by a large amount in the past 24 hours (e.g., words with a high "burstiness"). Moreover, they do not employ any personalization.

3.7 Conclusions

In this chapter, we set out to show that, in a basic setting with no queries and simple user interaction, our interactive concept coverage methodology can effectively solve a real world information retrieval problem that is not amenable to traditional keyword search. In order to demonstrate this, we gave a formal definition of a weighted concept representation and probabilistic notions of document coverage and set coverage. We defined a submodular objective function and described an efficient approximation algorithm for optimizing it to obtain a diverse set of relevant documents. Finally, as not all users have the same interests, we described an algorithm for learning user preferences for concepts from simple user feedback, achieving no regret versus the best fixed preference vector in hindsight.

On the application side, we showed that, for two different concept representations, our approach could successfully compete with or outperform state of the art commercial blog aggregation websites like Google Blog Search, Yahoo! Buzz, Digg, and BlogPulse. In addition to post content, most of these websites use richer features such as click-through rate, trends in search queries and link structure between posts, or use human intervention to pick posts. We present results based on simulations and a user study. Our TDN algorithm outperforms all others except for Yahoo! Buzz (with which it is comparable), despite having access to text-based features only. Furthermore, our experiments demonstrate that our algorithm can adapt to individual users' preferences.

Our results emphasize that the simple notion of coverage we introduced in our interactive concept coverage methodology successfully captures the salient stories of the day. We believe that this combination of coverage and personalization will prove to be a useful tool in the battle against information overload.

3.8 Appendix: No-Regret Learning

We cast our problem of learning a user's preferences in the framework of repeated matrix games. Each row i represents a concept. Our goal is to learn a probability distribution $\mathcal{P}$ over the concepts. Each column $\mathbf{f}^{(t)}$ represents the feedback for an ordered set of posts. The only difference from the Freund and Schapire framework [Freund and Schapire, 1999] is that our loss lies in the range $[-0.5, 0.5]$, instead of $[0, 1]$. This is because we define the loss for a cell $(i, \mathbf{f}^{(t)})$ in our matrix as,

$$\mathcal{L}(i, \mathbf{f}^{(t)}) := -\mathcal{M}(i, \mathbf{f}^{(t)}) = \frac{-w_i \sum_{d_j \in \mathcal{A}^{(t)}} f_j^{(t)} \;\; \textit{inc-cover}_j(d_{1:j-1}, i)}{2 \max_k w_k}.$$

The maximum value of $\sum_{d_j \in \mathcal{A}^{(t)}} f_j^{(t)} \;\; \textit{inc-cover}_j(d_{1:j-1}, i)$ is 1, and the minimum value is -1.

Let us denote the above game by $\mathcal{G}$. Consider another matrix game $\mathcal{G}'$ which has the same structure as $\mathcal{G}$. Define the loss function $\mathcal{L}'$ for the game $\mathcal{G}'$ as $\mathcal{L}'(i, \mathbf{f}^{(t)}) = \mathcal{L}(i, \mathbf{f}^{(t)}) + 0.5$. Thus, by construction, the loss function $\mathcal{L}'$ lies in the range $[0, 1]$. We will show that using multiplicative updates leads to a no-regret algorithm for the game $\mathcal{G}'$, and equivalently for the game $\mathcal{G}$.

Let the initial mixed strategy for game $\mathcal{G}'$ be $\mathcal{P}'_1$, and let $\mathcal{Q}'_t$ be the mixed strategy of the column player (i.e., the environment) at round t. After each round t, we compute a new mixed strategy for the next round using Freund and Schapire's multiplicative update rule:[12]

$$\mathcal{P}'_{t+1}(i) = \mathcal{P}'_t(i) \frac{\beta^{\mathcal{L}'(i, \mathcal{Q}'_t)}}{\mathcal{Z}'_t},$$

where $\mathcal{Z}'_t$ is the normalization factor, and $\beta \in [0, 1)$ is a parameter of the algorithm.

Similarly, let the initial mixed strategy for game $\mathcal{G}$ be $\mathcal{P}_1$. After each round t, we compute a new mixed strategy for the next round using the analogous multiplicative update rule:

$$\mathcal{P}_{t+1}(i) = \mathcal{P}_t(i) \frac{\beta^{\mathcal{L}(i, \mathcal{Q}_t)}}{\mathcal{Z}_t},$$

where $\mathcal{Z}_t$ is the normalization factor, and $\beta \in [0, 1)$ is a parameter of the algorithm. By definition, we see that,

$$\begin{aligned}
\mathcal{Z}_t &= \sum_{i=1}^{n} \mathcal{P}_t(i) \beta^{\mathcal{L}(i, \mathcal{Q}_t)} \\
&= \sum_{i=1}^{n} \mathcal{P}_t(i) \beta^{\mathcal{L}'(i, \mathcal{Q}'_t) - 0.5} \\
&= \frac{1}{\sqrt{\beta}} \sum_{i=1}^{n} \mathcal{P}_t(i) \beta^{\mathcal{L}'(i, \mathcal{Q}'_t)}.
\end{aligned}$$

[12] The reader should note that our original update equation (Eq. (3.7)) is defined directly in terms of $\mathcal{M}$, and thus contains a negative sign. We use loss in this proof to match the convention of Freund and Schapire.

Thus, if $\mathcal{P}_t(i) = \mathcal{P}'_t(i)$, then $\mathcal{Z}_t = \frac{1}{\sqrt{\beta}}\mathcal{Z}'_t$. Also, we notice that,

$$\begin{aligned}\mathcal{P}_{t+1}(i) =& \mathcal{P}_t(i)\frac{\beta^{\mathcal{L}(i,\mathcal{Q}'_t)}}{\mathcal{Z}_t} \\ =& \mathcal{P}_t(i)\frac{\beta^{\mathcal{L}'(i,\mathcal{Q}'_t)-0.5}}{\mathcal{Z}_t} \\ =& \frac{1}{\sqrt{\beta}}\mathcal{P}_t(i)\frac{\beta^{\mathcal{L}'(i,\mathcal{Q}'_t)}}{\mathcal{Z}_t}.\end{aligned}$$

If $\mathcal{P}_t(i) = \mathcal{P}'_t(i)$, then,

$$\begin{aligned}\mathcal{P}_{t+1}(i) =& \frac{1}{\sqrt{\beta}}\mathcal{P}'_t(i)\frac{\beta^{\mathcal{L}'(i,\mathcal{Q}'_t)}}{\frac{1}{\sqrt{\beta}}\mathcal{Z}'_t} \\ =& \mathcal{P}'_t(i)\frac{\beta^{\mathcal{L}'(i,\mathcal{Q}'_t)}}{\mathcal{Z}'_t} \\ =& \mathcal{P}'_{t+1}(i).\end{aligned}$$

Therefore, if we set $\mathcal{P}'_1 = \mathcal{P}_1$, then,

$$\forall t, \mathcal{P}'_t = \mathcal{P}_t. \tag{3.10}$$

We now use the following theorem from Freund and Schapire:

Theorem 3.8.1. *For any matrix $\mathcal{L}'$ with n rows and entries in $[0,1]$, and for any sequence of mixed strategies $\mathcal{Q}'_1, \ldots, \mathcal{Q}'_T$ played by the environment, the sequence of mixed strategies $\mathcal{P}'_1, \ldots, \mathcal{P}'_T$ produced by the multiplicative weights algorithm satisfies:*

$$\sum_{t=1}^{T}\mathcal{L}'(\mathcal{P}'_t, \mathcal{Q}'_t) \leq \min_{\mathcal{P}'}\left[\alpha_\beta \sum_{t=1}^{T}\mathcal{L}'(\mathcal{P}', \mathcal{Q}'_t) + c_\beta KL(\mathcal{P}'||\mathcal{P}'_1)\right]$$

$$\textit{where } \alpha_\beta = \frac{ln(1/\beta)}{1-\beta},\ c_\beta = \frac{1}{1-\beta}.$$

Now, suppose we set $\mathcal{P}'_1 = \mathcal{P}_1$. Then, as a corollary of the above theorem, we can say,

$$\begin{aligned}&\sum_{t=1}^{T}(\mathcal{L}(\mathcal{P}_t, \mathcal{Q}_t) + 0.5) \leq \min_{\mathcal{P}}\left[\alpha_\beta \sum_{t=1}^{T}(\mathcal{L}(\mathcal{P}, \mathcal{Q}_t) + 0.5) + c_\beta KL(\mathcal{P}||\mathcal{P}_1)\right] \\ \implies &\sum_{t=1}^{T}\mathcal{L}(\mathcal{P}_t, \mathcal{Q}_t) \leq \min_{\mathcal{P}}\left[\alpha_\beta \sum_{t=1}^{T}\mathcal{L}(\mathcal{P}, \mathcal{Q}_t) + c_\beta KL(\mathcal{P}||\mathcal{P}_1)\right] + 0.5\,T\,(\alpha_\beta - 1).\end{aligned}$$

If we set $\mathcal{P}'_1 = \mathcal{P}_1$ to the uniform distribution over the n rows of the matrix, we obtain,

$$\sum_{t=1}^{T}\mathcal{L}(\mathcal{P}_t, \mathcal{Q}_t) \leq \min_{\mathcal{P}}\left[\alpha_\beta \sum_{t=1}^{T}\mathcal{L}(\mathcal{P}, \mathcal{Q}_t) + c_\beta\, ln\, n)\right] + 0.5\,T\,(\alpha_\beta - 1). \tag{3.11}$$

Corollary 3.8.2. *If we set β to,*

$$\frac{1}{1+\sqrt{\frac{2\ ln\ n}{T}}},$$

the average per-trial loss suffered by the learner is,

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}(\mathcal{P}_t,\mathcal{Q}_t) \leq \min_{\mathcal{P}}\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}(\mathcal{P},\mathcal{Q}_t) + O\left(\sqrt{\frac{ln\ n}{T}}\right).$$

Proof. The right hand side of Eq. (3.11) consists of three components. We look at each one of them in turn. We first look at α_β.

For $\beta \in (0,1]$, by log series expansion we get,

$$\begin{aligned}
-ln\ \beta &= -2\sum_{n=0}^{\infty}\frac{1}{2n+1}\left(\frac{\beta-1}{\beta+1}\right)^{2n+1} \\
&= 2\sum_{n=0}^{\infty}\frac{1}{2n+1}\left(\frac{1-\beta}{1+\beta}\right)^{2n+1} \\
&\leq 2\sum_{n=0}^{\infty}\left(\frac{1-\beta}{1+\beta}\right)^{2n+1} \\
&= 2\left(\frac{1-\beta}{1+\beta}\right)\frac{1}{1-\left(\frac{1-\beta}{1+\beta}\right)^2} \\
&= 2\frac{(1-\beta)(1+\beta)}{(1+\beta)^2-(1-\beta)^2} \\
&= \frac{1-\beta^2}{2\beta}.
\end{aligned}$$

Therefore,

$$\alpha_\beta = \frac{-ln\ \beta}{1-\beta} \leq \frac{1+\beta}{2\beta} = \frac{1}{2\beta}+\frac{1}{2} = 1+\sqrt{\frac{ln\ n}{2T}}. \tag{3.12}$$

We now look at the second term in Eq. (3.11):

$$c_\beta ln\ n = \frac{1}{1-\beta}ln\ n = \left(1+\sqrt{\frac{T}{2ln\ n}}\right)ln\ n = ln\ n + \sqrt{\frac{Tln\ n}{2}}. \tag{3.13}$$

We now look at the third term in Eq. (3.11). By Eq. (3.12),

$$0.5T(\alpha_\beta - 1) \leq 0.5T\sqrt{\frac{ln\ n}{2T}}. \tag{3.14}$$

From Eq. (3.11), Eq. (3.12), Eq. (3.13), Eq. (3.14), and the fact that $\mathcal{L}(.,.) \leq 1$, we obtain,

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}(\mathcal{P}_t,\mathcal{Q}_t) &\leq \min_{\mathcal{P}}\frac{1}{T}\left[\alpha_\beta\sum_{t=1}^{T}\mathcal{L}(\mathcal{P},\mathcal{Q}_t)+c_\beta\, ln\; n)\right]+\frac{1}{T}0.5\,T\,(\alpha_\beta-1)\\
&\leq \min_{\mathcal{P}}\left[\left(\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}(\mathcal{P},\mathcal{Q}_t)+\sqrt{\frac{ln\; n}{2T}}\right)+\left(\frac{ln\; n}{T}+\frac{1}{T}\sqrt{\frac{Tln\; n}{2}}\right)\right]+0.5\sqrt{\frac{ln\; n}{2T}}\\
&= \min_{\mathcal{P}}\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}(\mathcal{P},\mathcal{Q}_t)+O\left(\sqrt{\frac{ln\; n}{T}}\right).
\end{aligned}$$

□

The inverse learning rate β depends on the number of rounds, T. As we often do not know this number in advance, one common approach, suggested by Freund and Schapire, is to divide the sequence into segments of increasing length, where the kth segment has length $T_k = k^2$. The learning rate for segment k can now be defined based on T_k, rather than the (unknown) total number of rounds, T. Freund and Schapire provide more details on the theoretical guarantees of this technique [Freund and Schapire, 1999].

3.9 Appendix: Data Preprocessing

In order to use the blog feeds gathered from Spinn3r, a series of preprocessing steps are first employed to clean the data. Spinn3r categorizes all posts as belonging to one of four categories: Weblog, Mainstream News, Forum, and Classified. We remove any posts that are tagged with the Forum or Classified labels, as we assume that users would not wish to be presented with such posts. This step reduces the number of posts by approximately half. Additionally, a short blacklist (121 sites) is used to filter out sites that are known to contain spam or that were misclassified as blogs. Finally, using a standard shingling approach, near-duplicate posts that appear *in the same blog* are removed. This allows us to deal with a common manifestation of post content extraction errors, as well as spam blog sites that often contain many duplicate posts. Note that we do not remove duplicate posts *across different blogs*, as these provide important information regarding the prevalence of a particular story.

At this point, the named entity recognizer and part of speech tagger are used to extract all the named entities and noun phrases, as described in sec3.4. After the standard steps of removing stop words and stemming the nouns, there are still over 20,000 total features. Retaining all of these features is wasteful, as most do not add any value, either because they rarely occur in the corpus or because they are uninformative for other reasons. Specifically, we perform the following feature selection steps on each eight hour epoch of data:

1. Select the 2,000 most frequent named entities, and discard the rest.
2. Select the 2,100 most frequent noun phrases, and discard the rest.[13] (The first two steps discard rarely occurring features.)
3. Discard the 200 most frequent noun phrases, as they tend to be uninformative (e.g., "Post" or "Comment").

[13] A noun phrase is defined to be a sequence of consecutive words all tagged as nouns by the part of speech tagger (i.e., "NN" or "NNS").

4. For each of the remaining noun phrases, compute how often it appears in each post in the corpus. Calculate the mean and variance of these counts for each noun phrase.

5. Use the statistics calculated in the previous step to prune away uninformative noun phrases. For instance, noun phrases with an extremely low variance of occurrence (i.e., whenever they occur in a document, they occur the same number of times), are often indicative of "boilerplate" text incorrectly parsed as post content (e.g., the word "copyright"). As another example, nouns that have a high average frequency of occurrence (i.e., whenever they occur, they occur many times) may indicate spam. Based on cursory observations, we empirically selected cutoff values[14] for these two statistics that allowed us to reduce the number of noun phrases we keep to below 1,000.

After this process, we are left with slightly fewer than 3,000 features per eight hour epoch, upon which we then run LDA.

3.10 Appendix: Setting the Concept Granularity Parameter

Following is a simple heuristic for appropriately setting the concept granularity parameter, ℓ, following the analysis presented in Figure 3.2.

// Define a redundancy threshold, indicating where along the horizontal axis in Figure 3.2 we want our coverage values to be.
$r \leftarrow 0.4$
// Compute the mean maximum concept probability for the document collection
$y \leftarrow \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \max_i P(c_i \mid d)$
if $y \geq r$ **then**
  $\ell \leftarrow 1$
else
  $\ell = \frac{\log(1-r)}{\log(1-y)}$

In words, this algorithm computes the ℓ required such that, on average, the most important concept per document achieves a level of coverage equal to the redundancy threshold.

[14] We keep noun phrases whose average frequency of occurrence in a post is in the range (1.3, 6), and whose variance is in the range (0.25, 12).

Chapter 4

Complex Queries and Trust Preferences

In the previous chapter, we examined how our interactive concept coverage approach can be successfully applied to address the problem of information overload in the blogosphere. Our methodology allowed us to elegantly formalize an important real-world problem, and, in turn, this problem setting allowed us to define and instantiate various steps of our methodology, such as probabilistic notions of document coverage and set coverage.

However, two steps of our methodology were not discussed in the previous chapter:

- While the blog and news recommendation setting is a queryless one, many domains exist where *queries are necessary*. Moreover, these queries may naturally be complex, not necessarily limited to traditional strings of keywords.
- The personalization that occurred in the previous chapter was focused on the content of the items being recommended. Specifically, we provided an algorithm for learning a personalized preference weighting over the concept set $\mathcal{C}$, via simple user feedback. However, we did not discuss how a *personalized affinity function* (Step 4 of our methodology) could be used to allow us to model user preferences that are orthogonal to item content, such as "I trust the *New York Times*."

In this chapter, we will show how our methodology can be extended to address these two challenges. We will do so by considering a domain where dealing with these challenges is critical to successfully addressing information overload: *discovering relevant scientific literature*.

4.1 Problem Description

Today, scientific researchers primarily rely on keyword search of online indices such as Google Scholar and PubMed. While these tools are indispensable, there are many instances where a researcher's information need cannot be easily specified as a simple string of keywords. Often, such a keyword query is either overly broad, returning many articles that are at best loosely related to the researcher's specific need, or too narrow, potentially returning no articles at all. In these occasions, it may be more natural for the scientist to specify his query as a small set of papers rather than as a set of words. In particular, having already read some articles that are related to the specific task at hand, the scientist can ask, "given that these papers represent my immediate research focus, what else should I read?".

More formally, given a small set of papers $\mathcal{Q}$ that we refer to as the *query set*, we seek to return a set $\mathcal{A}$ of additional papers that are related to the research focus defined by the query. Intuitively, a paper that cites all of the articles in $\mathcal{Q}$ is likely to represent related research. Likewise, a paper that is cited by every article in $\mathcal{Q}$ might contain relevant background information. However, it is restrictive to require the papers in $\mathcal{A}$ to have a direct citation to or from every article in the query set, as such papers are not guaranteed to exist. Instead, we wish to select a set $\mathcal{A}$ that maximizes a more general notion of *influence* to and from the papers in $\mathcal{Q}$.

Moreover, in scientific research, some articles command more respect than others, based on their authorship, publication venue, citation counts or other factors often unrelated to their textual content.

In the remainder of this chapter, we show how we can apply and extend our interactive concept coverage methodology to deal with these two problems:

1. We describe how we can extend our notion of *weighted concept representation* to incorporate the complex queries described above. We will do so by defining a document coverage function based on a notion of *influence* to and from the query documents.

2. We describe how we can learn a *personalized affinity function* over the documents in a corpus of scientific papers, modeling a user's trust preferences by using the aforementioned side information.

Finally, we present experimental results showing that researchers find the scientific papers recommended by our method to be more useful, trustworthy and diverse than those selected by popular alternatives, such as Google Scholar and a state-of-the-art topic modeling approach.

4.2 Modeling Scientific Influence

To define a notion of influence in scientific literature, we observe that the content of a publication is an amalgam of several sources, combining cited prior work with the authors' novel insights and background experience. For a given collection of articles, ideas travel *from cited papers to citing papers*, and from earlier to subsequent papers by the same author (Figure 4.1A). Our notion of influence should capture this transfer of ideas, modeling both the extent to which ideas travel between documents, as well as their topical matter. To achieve such fine-grained detail, we define influence with respect to the *individual concepts* found in a document collection. Just as in Chapter 3, these could be either coarse- or fine-grained features, such as topics, technical terms or key phrases. For example, we might say that the ideas transferred from one paper to another involve the concepts "energy" or "nitric oxide." For the remainder of this chapter, we will assume our set of concepts $\mathcal{C}$ to be a set of informative words extracted from the corpus.[1]

For each concept c in our vocabulary of concepts $\mathcal{C}$, we define a directed, acyclic graph G_c, where the nodes represent papers that contain c and the edges represent citations and common authorship.[2] Figures 4.1B and 4.1C show two such graphs for a subset of articles from the Proceedings of the National Academy of Sciences (PNAS), for the concepts "plant" and "stress." While a path between two nodes in such a graph may indicate influence with respect to a particular concept, mere existence of a path does little to express the *degree* to which this influence occurs. To capture the degree of influence, we define a weight $\theta^{(c)}_{x \to y}$ for each edge (x, y) in graph G_c, representing the probability of *direct* influence from paper x to paper y with

[1] More details can be found in Appendix 4.8, at the end of this chapter.

[2] In this chapter, we treat cited papers that are *also co-authored* simply as cited papers.

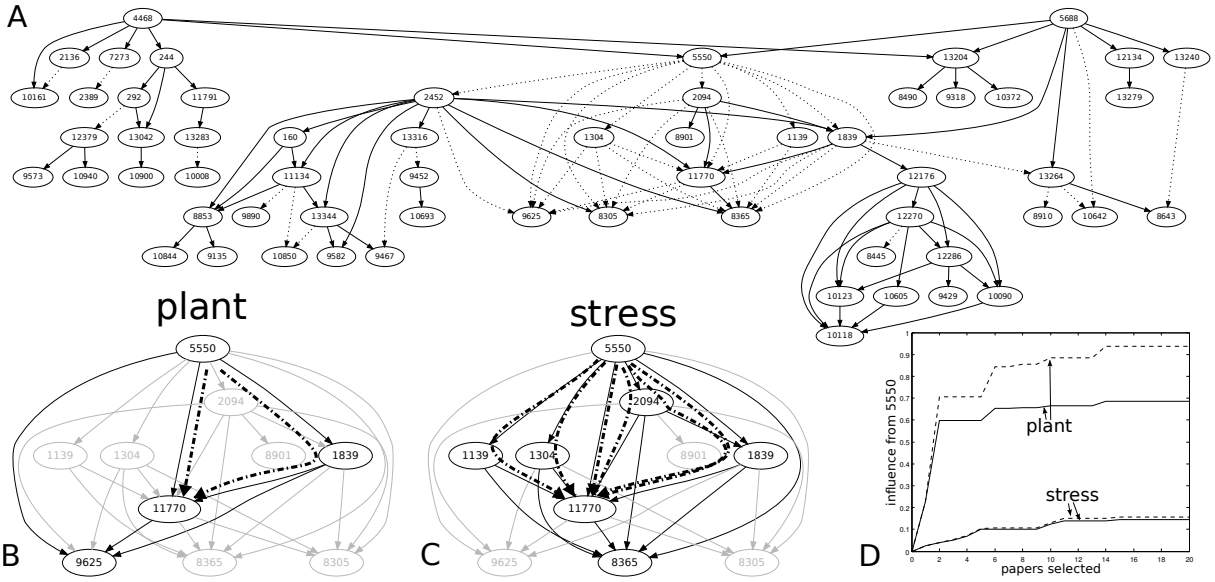


Figure 4.1: **(A)** A graph of articles from the Proceedings of the National Academy of Sciences (PNAS). Nodes represent papers, solid edges represent citations ($x \to y$ if y cites x) and dotted edges represent common authorship ($x \to y$ if x is older than y and x, y share an author). More details on the data sets used in this chapter can be found in Appendix 4.8. **(B,C)** Subgraphs of (A), limited to papers containing the concepts "plant" and "stress," respectively (other papers are grayed out). Thick dashed lines indicate paths of influence between papers 5550 and 11770. **(D)** Example illustrating how Eq. (4.2) penalizes redundancy. The first two papers selected exhibit a high influence with respect to "plant," and thus subsequently adding such papers to $\mathcal{A}$ causes little increase in Eq. (4.2) (solid lines), especially when compared to the sum of individual influences (dashed lines). The influence with respect to "stress" remains low, thus never triggering such a redundancy penalty.

respect to concept c. We can then use these edge weights to define a probabilistic, concept-specific notion of influence between any two papers in the document collection.

4.2.1 Defining edge weights

Figure 4.2 shows an example from the PNAS data set illustrating how we define the weight $\theta^{(c)}_{x \to y}$ on each edge. Here, article 9467 cites two articles containing the concept "oxygen," $\{424, 13344\}$, indicated by the solid black edges. The dotted black edges indicate that two other articles, $\{1829, 7657\}$, contain the concept "oxygen" and share authors with 9467. (The dotted gray edge indicates that there is a third article sharing authors with 9467 that *does not* contain "oxygen.") We assume that every occurrence of the concept "oxygen" in 9467 is either a novel idea or is directly inspired by one of these sources. Thus, we view the weight $\theta^{(c)}_{x \to y}$ as the probability a random instance of concept c in paper y was directly inspired by paper x.

The bar graph over the nodes in Figure 4.2 illustrates the proportion of the content of each paper consisting of the "oxygen" concept. For instance, the height of the first bar on the left is $n^{(oxygen)}_{424}/N_{424}$, where $n^{(c)}_x$ is the frequency of concept c in document x, and $N_x = \sum_{c \in \mathcal{C}} n^{(c)}_x$ is the total length of document x. Additionally, the bars over 1829 and 7657 are shortened to one third of their original height (indicated in light gray), representing the intuition that an explicit citation is a more informative relationship than common authorship. The authors of 9467 have three prior publications in this example, and thus by dividing

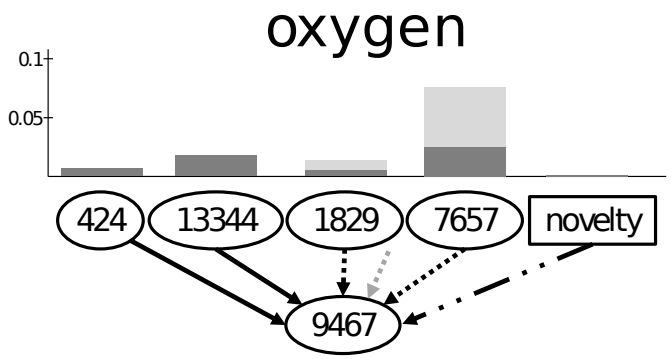


Figure 4.2: An example from the PNAS data set, illustrating the edge weight computation for a node in G_{oxygen}. Solid black edges indicate citations, while dotted black edges indicate common authorship. The dotted gray edge refers to a paper sharing an author with 9467, but not containing the concept "oxygen." Edge weights are assigned proportional to the bar chart, indicating the prevalence of "oxygen" in each parent node. The bars over 1829 and 7657 are shortened to one third of their original height (indicated in light gray), such that the contribution due to common authorship is equivalent to that of a single paper. The novelty node is only used to normalize the edge weights, and in this case is dominated in influence by the other articles.

by three, the effective total contribution of these papers is that of a single paper. Finally, we represent the novelty distribution for a particular paper y as the average distribution over concepts for all papers published in the same year as y. In this case, the novelty contribution for "oxygen" is dominated by the four papers. (We note that there are no actual novelty nodes in the graph, as the associated distribution is only used for normalization.)

Here, $\theta^{(oxygen)}_{x\to 9467}$ is proportional to the height of the corresponding bar in the plot. More generally, if a paper y cites papers $\{r_1, \ldots, r_k\}$, and the authors have previously written papers $\{b_1, \ldots, b_l\}$, then the edge weights are defined as follows:

$$\begin{aligned} \theta^{(c)}_{r_i\to y} &= \frac{1}{Z}\frac{n^{(c)}_{r_i}}{N_{r_i}}, \\ \theta^{(c)}_{b_i\to y} &= \frac{1}{Z}\frac{n^{(c)}_{b_i}}{l\cdot N_{b_i}}, \end{aligned}$$

with normalization constant,

$$Z = \sum_{j=1}^{k}\frac{n^{(c)}_{r_j}}{N_{r_j}} + \frac{1}{l}\sum_{j=1}^{l}\frac{n^{(c)}_{b_j}}{N_{b_j}} + novel^{(c)}_y,$$

where $novel^{(c)}_y$ is the average proportion of concept c across all papers published in the same year as y.

4.2.2 Calculating influence

Given a concept-specific weight for each edge in the citation graph, representing the *direct* influence between two neighboring nodes, we can now define the influence between any two papers in our collection. In particular, if we say that each edge $x \to y$ in G_c is *active* with some probability $\theta^{(c)}_{x\to y}$, we arrive at the following definition:

Definition 4.2.1. *The influence between papers u and v with respect to concept c, $Influence_c(u \leftrightarrow v)$, is the probability there exists a directed path in G_c from one paper to the other, consisting only of active edges.*

While intuitive, the exact computation of this probability is intractable, as the problem of computing connectedness in a random graph belongs to the #P-complete class of computational problems [Provan and Ball, 1983, Valiant, 1979], for which there are no known polynomial-time solutions. We can overcome this computational hurdle via approximation, by employing one of two methods: 1) a simple Monte Carlo sampling procedure with theoretical guarantees; and, 2) a deterministic, linear-time dynamic programming heuristic, based on the assumption that the paths between two nodes are independent of each other.

Sampling

The simplest procedure for estimating the influence between two nodes is to generate samples directly based on the definition of influence. Each sample is generated as follows:

For each concept c:

1. Mark each edge $x \rightarrow y$ in G_c as active with probability $\theta^{(c)}_{x \rightarrow y}$.

2. For all pairs of nodes (u, v), record whether a path exists between them using only active edges.

After generating B samples, the probability that a node u influences a node v with respect to concept c is simply estimated as the proportion of the B samples for concept c in which an active path from u to v exists. A natural question to ask is, how many samples do we need for a reasonable estimate of influence? A short proof using Hoeffding's Inequality shows us that the number of samples we need grows only *logarithmically* with the number of articles in the document collection.

Theorem 4.2.1. *In order to estimate m influence values such that, with probability η,* each *of the m estimates is no more than δ away from its true value, a sufficient number of samples B is $\frac{2}{\delta^2}\log(2m/\eta)$.*

Proof. We wish to estimate m influence probabilities, $p_1, p_2, \ldots, p_m$, using $\widehat{p}_1, \widehat{p}_2, \ldots, \widehat{p}_m$, where $\widehat{p}_j = \frac{1}{B}\sum_{i=1}^{B} X_j^{(i)}$, and each $X_j^{(i)}$ is a random variable that is either 0 or 1, representing whether the jth pair of nodes is connected via an active path in sample i. Note that by our definition of influence, $E[\widehat{p}_j] = \frac{1}{B}\sum_{i=1}^{B} E[X_j^{(i)}] = p_j$.

We let $\epsilon_j = |\widehat{p}_j - p_j|$, the absolute difference between influence value p_j and its estimate using the sampling

methodology from above. Given some δ, we want $P(\epsilon_1 \geq \delta \vee \epsilon_2 \geq \delta \vee \ldots \vee \epsilon_m \geq \delta)$ to be small.

$$\begin{aligned}
P\left(\bigvee_{j=1}^{m}(\epsilon_j \geq \delta)\right) &\leq \sum_{j=1}^{m} P\left(\epsilon_j \geq \delta\right) \\
&= \sum_{j=1}^{m} P\left(|\widehat{p}_j - p_j| \geq \delta\right) \\
&= \sum_{j=1}^{m} P\left(\left|\frac{1}{B}\sum_{i=1}^{B} X_j^{(i)} - \frac{1}{B}\sum_{i=1}^{B} E\left[X_j^{(i)}\right]\right| \geq \delta\right) \\
&= \sum_{j=1}^{m} P\left(\left|\frac{1}{B}\sum_{i=1}^{B} X_j^{(i)} - E\left[X_j^{(i)}\right]\right| \geq \delta\right) \\
&= \sum_{j=1}^{m} P\left(\left|\sum_{i=1}^{B} X_j^{(i)} - B \cdot E\left[X_j^{(i)}\right]\right| \geq B\delta\right) \\
&\leq \sum_{j=1}^{m} 2\exp\left(\frac{-2B^2\delta^2}{4B}\right) \\
&= 2m\exp\left(\frac{-B\delta^2}{2}\right),
\end{aligned}$$

where the first inequality is due to the union bound, and the second inequality is due to Hoeffding [1963].

Thus, the probability that any of our m estimates is more than δ away from its true value given B samples is less than or equal to $2m\exp(-B\delta^2/2)$. For this probability to be less than or equal to η, we need:

$$\begin{aligned}
2m\exp\left(\frac{-B\delta^2}{2}\right) &\leq \eta, \\
\exp\left(\frac{-B\delta^2}{2}\right) &\leq \frac{\eta}{2m}, \\
-B\delta^2 &\leq 2\log\left(\frac{\eta}{2m}\right), \\
B &\geq \frac{-2}{\delta^2}\log\left(\frac{\eta}{2m}\right) \\
&= \frac{2}{\delta^2}\log\left(\frac{2m}{\eta}\right).
\end{aligned}$$

□

As the number of influence values to estimate is quadratic in the number of articles, the number of samples we need is logarithmic in the total number of articles. While this is a heartening result, we find that for large document collections, generating enough samples can still be a time-consuming process.

Independence heuristic

As an alternative to sampling, we describe an efficient dynamic programming heuristic based on the assumption that the paths between two nodes in G_c are independent of each other. For instance, in Figure

1B, the two influence paths between 5550 and 11770 with respect to the concept "plant" are completely independent of each other. Thus, the probability of at least one active path existing between the two nodes in this situation can be computed exactly:

$$\begin{aligned}
&\mathit{Influence}_{plant}(5550 \to 11770) \\
&\quad = \quad 1 - P(\text{there is no influence between 5550 and 11770}) \\
&\quad = \quad 1 - P(\text{there is no direct influence from 5550}) \cdot \\
&\qquad\quad P(\text{there is no influence through 1839}) \\
&\quad = \quad 1 - (1 - \theta^{(plant)}_{5550 \to 11770})(1 - \theta^{(plant)}_{5550 \to 1839}\theta^{(plant)}_{1839 \to 11770}).
\end{aligned}$$

The second equality follows from the independence of the two paths. On the other hand, looking at Figure 1C, we find the paths between the two nodes in G_{stress} are not independent, making such a calculation more problematic.

Based on this intuition, if we rashly assume that the paths between two nodes will *always* be independent of each other in G_c, for all c, we arrive at a simple, efficient heuristic for computing the influence between all pairs of nodes (Algorithm 4.1). By traversing the graph in topological order, we know that when we arrive at a node we will have already computed all the influence going to its parents. Using these influences and our independence assumption, we can then immediately compute the influence to the node itself. We note that this algorithm requires the graphs to be acyclic.[3]

While the independence assumption upon which this heuristic is based certainly is not true in general, we find that, nevertheless, the values we compute are close to what we would expect from sampling (cf. Figure 4.3). Thus, despite not being amenable to theoretical guarantees, we find this heuristic works well in practice.

4.3 Selecting Articles

Thus far, we have defined a probabilistic notion of influence between any two documents in a corpus of research papers. However, our end goal is to select a set of relevant articles that exhibit high influence to or from a set of query papers. In this section, we show how we can apply our interactive concept coverage methodology to address this problem. At a high level, we will maintain the same set coverage function and overall objective function as defined in the previous chapter, while defining a new concept representation designed to take the query set of documents into account.

We start by defining an *augmented concept representation*:

Definition 4.3.1 (Augmented Concept Representation)**.** *An* augmented concept representation *is a quintuple* $\langle \mathcal{C}, \mathcal{D}, \mathcal{Q}, \mathit{cover}_{\cdot}(\cdot,\cdot), \mathbf{w} \rangle$*.* $\mathcal{C}$ *is a finite set of concepts,* $\mathcal{D}$ *is a finite set of documents and* $\mathcal{Q}$ *is a finite set of* augmented items*. The relation between documents, concepts and augmented items is captured by the* augmented document coverage function*,* $\mathit{cover}_j(i,q) : \mathcal{C} \times \mathcal{Q} \to \mathbb{R}_{\geq 0}$*, which quantifies the amount document* $d_j \in \mathcal{D}$ *covers concept* $c_i \in \mathcal{C}$ *with respect to item* $q \in \mathcal{Q}$*.* $\mathbf{w} \in \mathbb{R}_{\geq 0}^{|\mathcal{C}| \times |\mathcal{Q}|}$ *is a weight vector over concept-item pairs, indicating relative importance.*

[3] Based on simple chronology, one would expect a citation graph to be acyclic; after all, a researcher cannot cite a paper if it does not yet exist. However, this is not quite the case in practice (e.g., colleagues writing papers simultaneously may cite each other). Details on how we address this problem can be found in Appendix 4.8, at the end of this chapter.

Algorithm 4.1: Dynamic Programming Heuristic for Influence

N: number of documents
$\mathcal{C}$: vocabulary of concepts
// Initialize to empty 3D array
// $influenceEstimate[c][x][y]$ will contain influence
// from x to y with respect to concept c.
$influenceEstimate \leftarrow array[|\mathcal{C}|][N][N]$
for all $c \in \mathcal{C}$ **do**
  for all nodes y in G_c **do**
    // Initialize to identity
    $influenceEstimate[c][y][y] \leftarrow 1$
  $topoOrder \leftarrow$ topological order of nodes in G_c
  for $y \in topoOrder$ **do**
    // $influenceEstimate[c][][x]$ already calculated
    // for all $x \in parents(y)$
    if $parents(y) = \emptyset$ **then**
      continue
    $influenceFromParents \leftarrow array[|parents(y)|]$
    for all $x \in parents(y)$ **do**
      // Influence to the parent multiplied by
      // the edge weight
      $influenceFromParents[x] \leftarrow influenceEstimate[c][][x] \cdot \theta^{(c)}_{x \to y}$
    // Product is element-wise
    $influenceEstimate[c][][y] \leftarrow 1 - \prod_{x \in parents(y)} (1 - influenceFromParents[x])$

4.3.1 Influence-based Coverage

Rather than simply covering concepts, we will use the augmented concept representation to define a document coverage function over concept-query document pairs. Intuitively, we want an article d to cover a concept c with respect to query document q if:

- There is a high degree of influence between documents d and q with respect to concept c; and,
- Concept c is important in document d.

The first condition reiterates the initial motivation of this chapter, of finding articles with high influence to and from the query set, and can be satisfied by ensuring that our document coverage function includes our expression for influence: $\mathit{Influence}_c(q \leftrightarrow d)$.

The second condition ensures that a document cannot cover a concept unless that concept is prominently featured in that document. For example, a document d might contain a single occurrence of the concept "plant," and that single occurrence might be heavily influenced by one of the query documents q. However, as d only tangentially mentions "plant," we do not wish this strong influence to incentivize its inclusion in the result set. To address this concern, we again assume we have a probability distribution over concepts for document d, $P(c \mid d)$, and a concept granularity parameter ℓ (cf. Section 3.1), which we can use to define

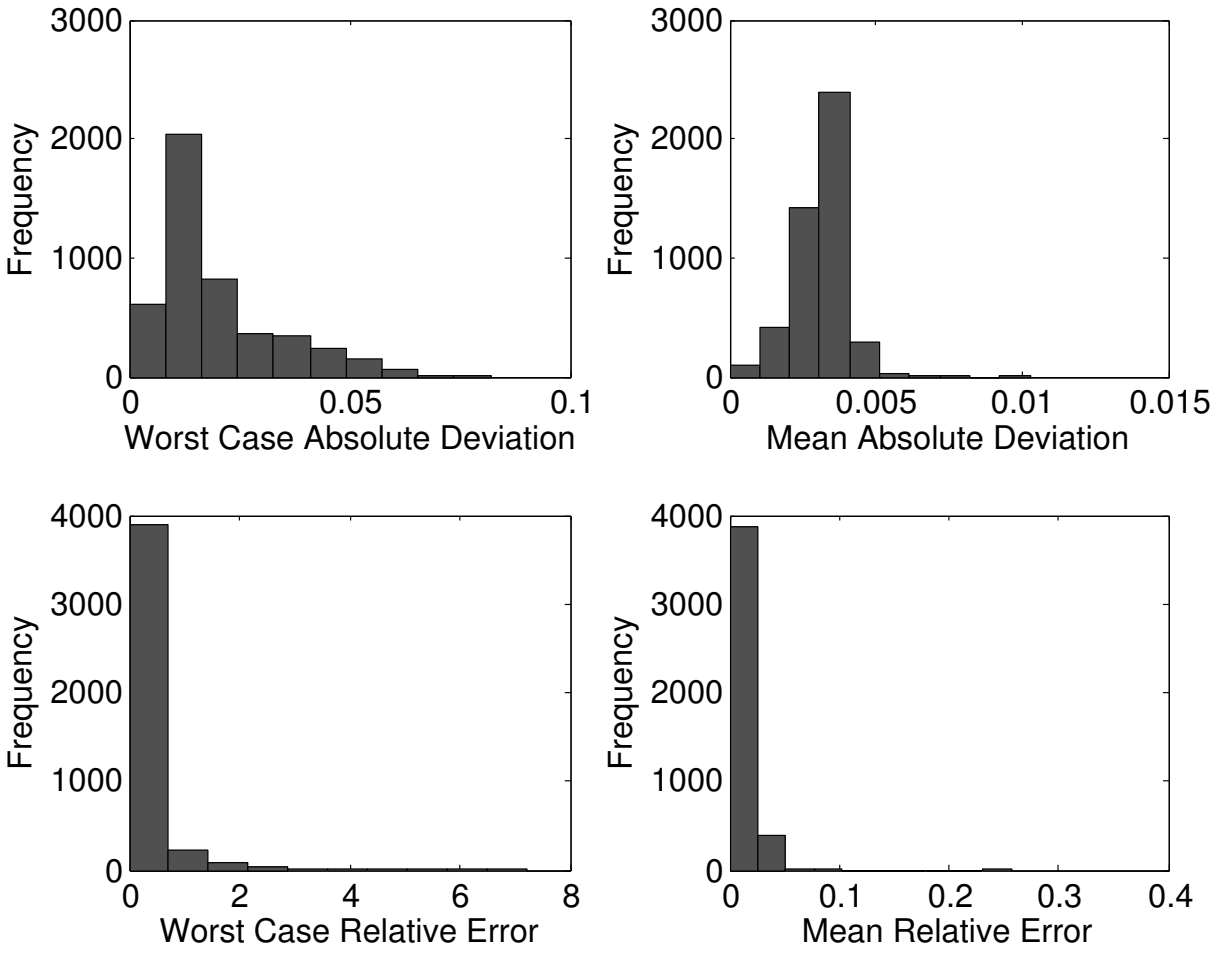


Figure 4.3: This figure shows a comparison on the PNAS data set between the influence values computed via sampling ($B = 6530$) and those computed using the independence heuristic. For all concepts and all pairs of articles with meaningful influence between them (i.e., not trivially zero, as is the case when the nodes are not connected in the graph), we compute the influence using both methods, and record the absolute deviation ($|\textit{sampling} - \textit{heuristic}|$) and relative error ($|\textit{sampling} - \textit{heuristic}|/\textit{sampling}$). The worst case and mean values of these measures for each concept are plotted above. For this setting of B, the estimates computed via sampling are likely ($> 95\%$) to be within 0.075 of their true values.

an importance probability $\beta_d^{(c)} = 1 - (1 - P(c \mid d))^{\ell}$.[4] We note that this is identical to the expression for probabilistic document coverage in Eq. (3.1). We use this importance probability to safeguard against selecting documents that are only tangentially related to the imported concepts in the query papers.

Putting these together, we obtain the following augmented document coverage function:

$$\textit{cover}_d(c, q) = \textit{Influence}_c(q \leftrightarrow d) \cdot \beta_d^{(c)}. \tag{4.1}$$

Finally, to ensure that the selected documents pertain to the concepts most prevalent in the query set, we define the concept-item weight vector $\mathbf{w}$ such that $w_{c,q}$ is proportional to the frequency of concept c in

[4]For the experiments in this chapter, we set the concept granularity parameter $\ell = 20$ and $P(c \mid d)$ to be proportional to the frequency of concept c in document d.

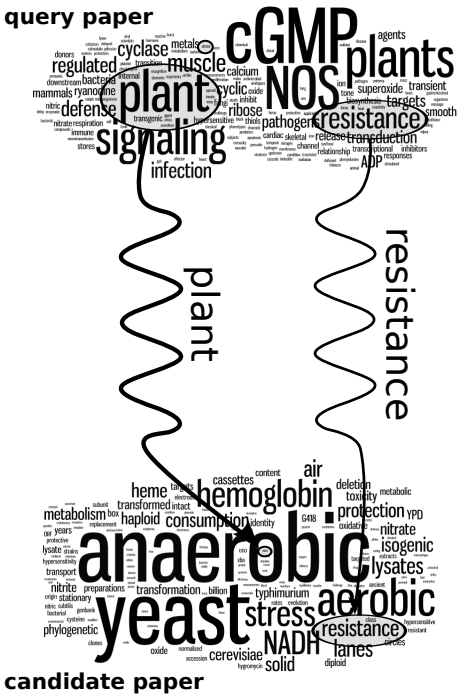


Figure 4.4: The top cloud represents a query paper (5550), the bottom word cloud represents a paper to be selected (11770) and the lines between them represent individual influences of varying strength. In each word cloud, the size of a word is proportional to its frequency in the corresponding article. $\mathbf{w}$ is illustrated by the shaded ellipses in the top word cloud, showing a higher incentive to pick articles about "plant" or "resistance" than about "stress." However, despite its prevalence in the query document, "plant" is only tangentially present in article 11770, and thus β ensures a low degree of influence. This can be contrasted with "resistance," which is prevalent in both documents and displays a high degree of influence.

query document q.

Figure 4.4 provides an illustrative example of the weights defined in this section.

4.3.2 Optimization

Given the augmented concept representation we defined above, we can now extend the probabilistic set coverage function and objective from the previous chapter to allow us to select a diverse set of scientific articles, related to the query set. We note that diversity is important in this setting as it is difficult to predict the exact information need of a researcher, and thus providing a wide variety of papers increases the likelihood of query satisfaction.

Our augmented probabilistic set coverage function is:

$$cover_{\mathcal{A}}(c, q) = 1 - \prod_{d \in \mathcal{A}} (1 - cover_d(c, q)). \tag{4.2}$$

This is the natural extension of our set coverage function from the last chapter, in Eq. (3.2), to the augmented case, indicating that a set of documents $\mathcal{A}$ covers concept c with respect to query article q if at least one of the articles in $\mathcal{A}$ covers c with respect to q. The diminishing returns property exhibited by this function is

illustrated in Figure 4.1D, showing how the marginal gain in augmented set coverage of the concept "plant" diminishes as more papers are added to the result set $\mathcal{A}$. In particular, beyond a certain level of influence, the gain observed in Eq. (4.2) from adding additional documents to the result set is smaller than would be expected if we were naïvely summing the individual influences. We do not see the same redundancy penalty with respect to "stress," as the result set is not sufficiently influenced with respect to this concept.

Finally, we can put everything together by extending our submodular objective function from Chapter 3 to this setting, such that, when maximized, it returns a diverse set of papers highly relevant to the query:

$$F_{\mathcal{Q}}(\mathcal{A}) = \sum_{q \in \mathcal{Q}} \sum_{c \in \mathcal{C}} w_{c,q} \; cover_{\mathcal{A}}(c, q). \tag{4.3}$$

This objective is of identical form to Eq. (3.3) in the previous chapter, has the same theoretical guarantees, and can be optimized using the same efficient CELF algorithm [Leskovec et al., 2007].

4.4 Trust and Personalization

Considering our running example of PNAS articles (Figure 4.1A), we can set our query set to be $\mathcal{Q} = \{4468, 5688\}$, the parents of "Nitric Oxide in Plant Immunity" (5550). Optimizing Equation 4.3 for this query produces a result set of articles ranging in topics from plant biology to immunology (cf. Table 4.2). While these articles may be relevant to the query, a major shortcoming is that every researcher who submits this query will receive an identical result set. For any given topic, different researchers trust different authors and publications, and the objective in Equation 4.3 provides no means to express these preferences. Unlike the content-based personalization introduced in Chapter 3, the personal preferences we seek to model in this section are based on side information of the documents being recommended, such as authorship or publication venue. As such, we introduce a *personalized affinity function*, defined over documents, to encode such preferences.

While a long line of prior work exists on summarizing the impact of an author or publication with a single number [Adler et al., 2009], often based on citation statistics [Garfield, 1972, Hirsch, 2005] or eigenvector methods [Chen et al., 2007, Kleinberg, 1999, Page et al., 1999, Radicchi et al., 2009], here we wish to capture a more detailed picture of the relationship between a researcher and the authors he cites.

In order to properly model such an individual notion of *trust* in the setting of scholarly research, we consider two motivating scenarios:

1. Different authors command different levels of respect from their research communities, e.g., a Nobel laureate versus a first-year graduate student, as an extreme case.

2. Even among distinguished scientists, a particular researcher's interests may be aligned more closely with some than others. Thus, beyond simply differentiating novices from experts, a notion of trust should also capture differences in research interests. For example, asking computer scientists to name whom they most associate with the concept "network" may yield Judea Pearl (Bayesian networks), Jon Kleinberg (social networks), Geoff Hinton (neural networks) or Van Jacobson (computer networks), depending on who is answering. All are distinguished researchers, but each is associated with a distinct subfield of computer science.

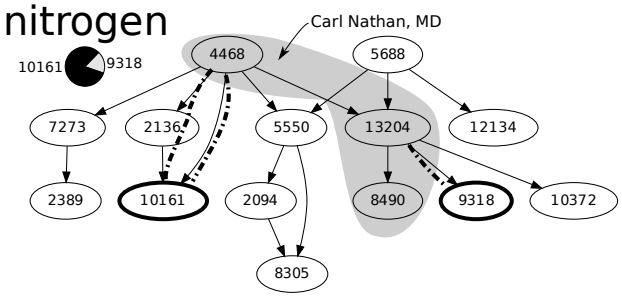


Figure 4.5: Example illustrating trust calculation for an immunologist asking, "How much do I trust Carl Nathan with respect to the concept 'nitrogen'?" Thick dashed lines indicate influence from Dr. Nathan to individual elements of $\mathcal{B}$, and pie chart represents relative prevalence of the word "nitrogen" in the two papers in $\mathcal{B}$.

At the heart of both scenarios is a personal question that is often answered differently by different researchers: *How much do I trust this author with respect to this concept?*

By answering this question, a researcher would enable us to formally incorporate his trust preferences into our objective function, allowing us to select papers tailored specifically to his tastes. However, as researchers will not be able to provide an answer for every combination of authors and concepts, we must elicit their trust preferences in a less onerous manner. In order to do so, we assume that trust is *transitive*. For example, if Alice trusts an article, and that article is heavily influenced by Bob with respect to the concept "network," then Alice is likely to also trust Bob with respect to "network." Thus, at a fundamental level, a researcher need only specify a set of trusted papers $\mathcal{B}$, from which we can infer answers to the above question. As a shortcut, a researcher may choose to define $\mathcal{B}$ indirectly by specifying a list of trusted journals and conferences, or subsets thereof (e.g., a particular conference track or article classification). $\mathcal{B}$ could also be specified as the papers *cited by* one or more trusted authors, representing a look at one's research through the eyes of another scientist, potentially in another field. Thus, a plucky physicist could ask, "What would Steven Chu recommend I read?", and obtain a set of papers related to his query, yet tailored to the research interests and trust preferences of the Nobel laureate.

With this intuition in mind, we define $\tau^{(c)}_{a|\mathcal{B}}$, the probability a researcher trusts author a with respect to concept c, given trusted articles $\mathcal{B}$. (The "$|\mathcal{B}$" notation in this section indicates personalizing with respect to trusted set $\mathcal{B}$.) Figure 4.5 illustrates how we compute $\tau^{(c)}_{a|\mathcal{B}}$ for a particular example from PNAS, where the concept c is "nitrogen," the author a is Carl Nathan, MD, and the researcher has specified two immunology papers as his trusted set, $\mathcal{B} = \{10161, 9318\}$. For each paper $b \in \mathcal{B}$, we compute how much the author a influenced b with respect to concept c. As our influence is now expressed from an *author* to an article, we treat all of an author's papers as a single unit.

Definition 4.4.1. *The influence from author a to article b with respect to concept c, $AuthorInfluence_c(a \to b)$, is the probability there exists a directed path in G_c from* any *article written by a to article b consisting only of active edges, where each edge is (independently) active with probability $\theta^{(c)}_{x \to y}$.*

As before, we employ sampling or dynamic programming to efficiently estimate this otherwise intractable computation (cf. Algorithm 4.2).

In our example, we first look at how much Dr. Nathan's papers influence 10161 with respect to "nitrogen," and again from Dr. Nathan's papers to 9318. We now weigh these two influences by the prevalence of

Algorithm 4.2: Dynamic Programming Heuristic for Author Influence

N: number of documents
$\mathcal{C}$: vocabulary of concepts
// Initialize to empty 3D array
// $authorInfluence[c][a][y]$ will contain influence from author a to paper y w.r.t. concept c.
$authorInfluence \leftarrow array[|\mathcal{C}|][numAuthors][N]$
for all $c \in \mathcal{C}$ **do**
  for all authors a **do**
    // Every author influences his or her own papers
    $authorInfluence[c][a][papers(a)] \leftarrow 1$
  $topoOrder \leftarrow$ topological order of nodes in G_c
  for $y \in topoOrder$ **do**
    // $authorInfluence[c][][x]$ already calculated for all $x \in parents(y)$
    if $parents(y) = \emptyset$ **then**
      continue
    $influenceFromParents \leftarrow array[|parents(y)|]$
    for all $x \in parents(y)$ **do**
      // Influence to the parent multiplied by the edge weight
      $influenceFromParents[x] \leftarrow authorInfluence[c][][x] \cdot \theta^{(c)}_{x \to y}$
    // Product is element-wise
    $authorInfluence[c][][y] \leftarrow 1 - \prod_{x \in parents(y)}(1 - influenceFromParents[x])$
    // Retain authors' self-influence
    $authorInfluence[c][authors(y)][y] \leftarrow 1$

the word "nitrogen" in each paper b (as indicated by the pie chart in Figure 4.5), and define $\tau^{(c)}_{a|\mathcal{B}}$ to be the weighted sum of the two.

More generally, we have:

$$\tau^{(c)}_{a|\mathcal{B}} = \begin{cases} \frac{1}{N^{(c)}_{\mathcal{B}}} \sum_{b \in \mathcal{B}} n^{(c)}_b \, AuthorInfluence_c(a \to b), & \text{if } N^{(c)}_{\mathcal{B}} > 0 \\ \tau^{(c)}_{a|\mathcal{D}}, & \text{otherwise,} \end{cases}$$

where $N^{(c)}_{\mathcal{B}}$ is the total weight of concept c in the set $\mathcal{B}$, $n^{(c)}_b$ is the weight of concept c in paper b, and $\mathcal{D}$ is the set of all papers in the corpus. Here, the influence to each $b \in \mathcal{B}$ is weighted by the relative prevalence of concept c with respect to $\mathcal{B}$, $n^{(c)}_b / N^{(c)}_{\mathcal{B}}$. We note that if a researcher's trusted set $\mathcal{B}$ contains no occurrences of a particular concept, we assign the trust value to $\tau^{(c)}_{a|\mathcal{D}}$, as if all the papers in the corpus were trusted equally.

In order to incorporate trust into paper selection, we assume an author will trust a paper if and only if he trusts *at least one of its authors*. This intuition can be encoded in a personalized affinity function:

$$\textit{affinity}(d) = 1 - \prod_{a \in authors(d)} (1 - \tau^{(c)}_{a|\mathcal{B}}).$$

We can now include this affinity into our augmented document coverage function, thereby encoding the

personal trust preferences of the user:

$$\textit{cover}_{d|\mathcal{B}}(c, q) = \textit{Influence}_c(q \leftrightarrow d) \cdot \beta_d^{(c)} \cdot \textit{affinity}(d). \tag{4.4}$$

We can plug this in to the same objective function as in the unpersonalized case—Eq. (4.3)—and optimize it the same way with the same guarantees. The objective remains submodular and monotonic, as the document coverage function is still defined as a probability, over the range $[0, 1]$.

Figure 4.6 shows our PNAS example from before, with the same query set $\mathcal{Q} = \{4468, 5688\}$, but now incorporating the trust preferences of two hypothetical researchers: a plant biologist (A) and an immunologist (B). The figure highlights how differences in trust preferences can manifest themselves in article selection. In Figure 4.7, we provide another example, this time from computer science. Here, we take the famous Faloutsos, Faloutsos and Faloutsos paper, "On power-law relationships of the Internet topology" [Faloutsos et al., 1999], and select related literature for it using the trust preferences of each author. Specifically, the visualization in the figure shows that by assuming that Michalis Faloutsos trusts SIGCOMM papers, Petros Faloutsos trusts SIGGRAPH papers, and Christos Faloutsos trusts KDD papers, we can select related work tailored to each author's perspective. While some relevant papers are common to all three points of view, other selected papers are particular to just one. For example, in Christos' data mining-focused result set, we find a few papers related to the evolution of social networks (e.g., "Microscopic evolution of social networks" by Leskovec et al.) which are not found in Michalis' and Petros' results. Moreover, these papers are not selected in the unpersonalized setting, when no trust preferences are taken into account.

4.5 Experimental Results

While these illustrative examples provide intuition, in order to truly evaluate our methodology we must solicit feedback from real scientific researchers. To this end, we conducted a user study involving sixteen subjects (all doctoral students in computer science or related fields).

We compare two variants of our algorithm–with and without incorporating trust preferences of the participant–with three representative alternative techniques: Google Scholar,[5] Information Genealogy [Shaparenko and Joachims, 2007] (a hypothesis testing approach based on document text), and the Relational Topic Model [Chang and Blei, 2010] (a state-of-the-art topic model incorporating both text and citations to model latent themes in data).[6] For each participant, we use each of these techniques to find related work for a previously written paper–that participant's *study paper*–thereby simulating a real research scenario. We define each query set $\mathcal{Q}$ to be the references of the corresponding study paper, and we ask each participant to list up to four trusted conferences or journals, which we use to define $\mathcal{B}$. The articles used in this study come from the ACM Digital Library,[7] as described in Appendix 4.8.

In the case of Google Scholar, we ask a coauthor of the participant to provide the ideal keyword query he or she would use to find related work for the study paper. We enter this query into Google Scholar, and retrieve a result set containing the top ten papers that also appear in our ACM data set. In some cases, the keyword query provided was too specific, resulting in fewer than ten Google Scholar results.

[5] `http://scholar.google.com`

[6] Previous work [McNee et al., 2002] has shown that Google keyword search outperforms collaborative filtering techniques for selecting useful papers, and thus we do not directly compare against these approaches.

[7] `http://portal.acm.org`

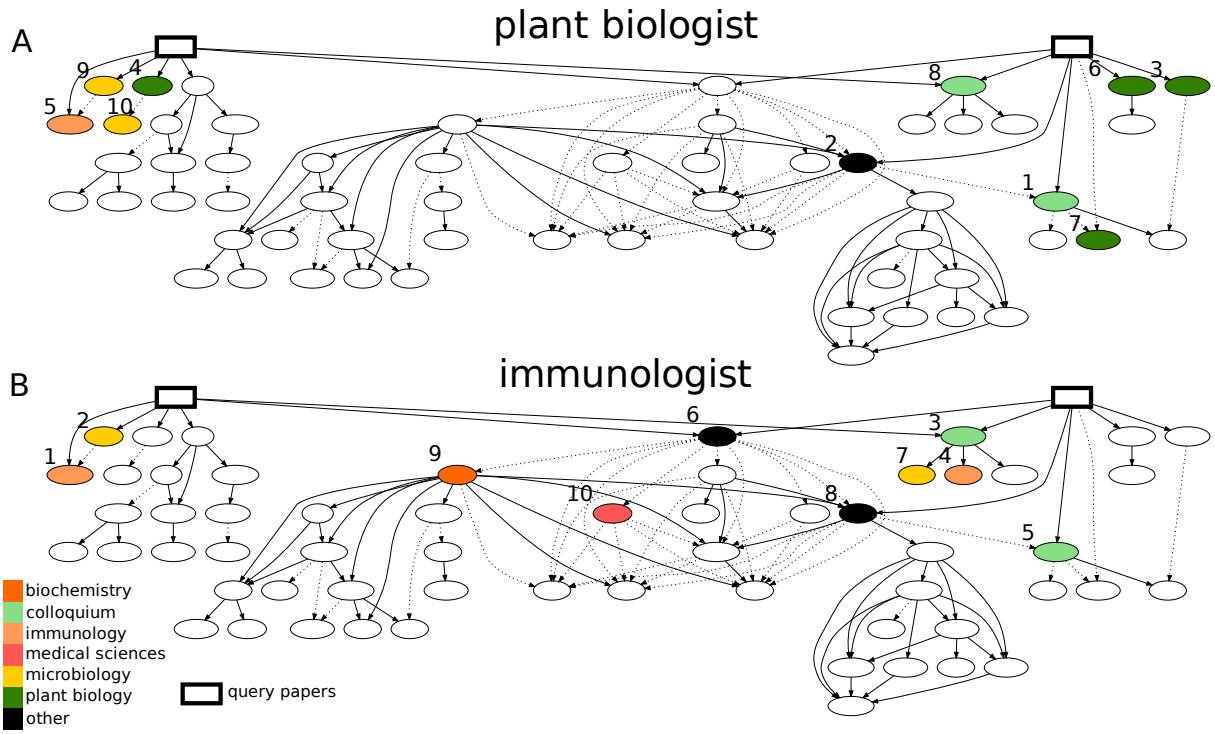


Figure 4.6: Top ten papers selected for $\mathcal{Q} = \{4468, 5688\}$ where $\mathcal{B}$ is defined as (A) all the plant biology papers, or (B) all the immunology papers, in the PNAS data set. Node colors correspond to article classification, as indicated by the key. (Colloquium refers to the National Academy of Sciences Colloquium on Virulence and Defense in Host-Pathogen Interactions: Common Features Between Plants and Animals. "Other" refers to unclassified papers, e.g., "From the Academy.".) Numbers indicate order of selection by optimization algorithm, roughly indicating order of importance (cf. Tables 4.3 and 4.4).

For the Relational Topic Modeling approach, we fit the model to our data using the collapsed Gibbs sampling package provided by the authors [Chang, 2010]. We use K=50 topics, a burn-in of 750 samples, and collect our results over 750 additional samples. The parameters are set according to guidance from the first author (alpha=1/K, beta=4, eta=1/(size of vocabulary)). To select a set of related work, we compute the link probability from the study paper to each additional paper, and return the top ten most likely new links. We note that we give the model access to the abstract of the study paper–information that our algorithms do not have access to.

Finally, as the Information Genealogy model only takes into account document text, we provide the algorithm with the abstract of the study paper and retrieve the ten papers in the corpus with the most influence on the study paper. We use the same convex optimization package as used by the authors of the paper.[8]

Unlike many previous studies, each participant was asked to evaluate all five comparison methods, rather than just a single technique. In total, 612 distinct papers were recommended using these five techniques across all sixteen participants.

Each participant was presented with the recommended articles for his or her study paper in a double-blind fashion, masking the identity of the technique used to select each paper. Participants were asked

[8] `http://www.mosek.com`

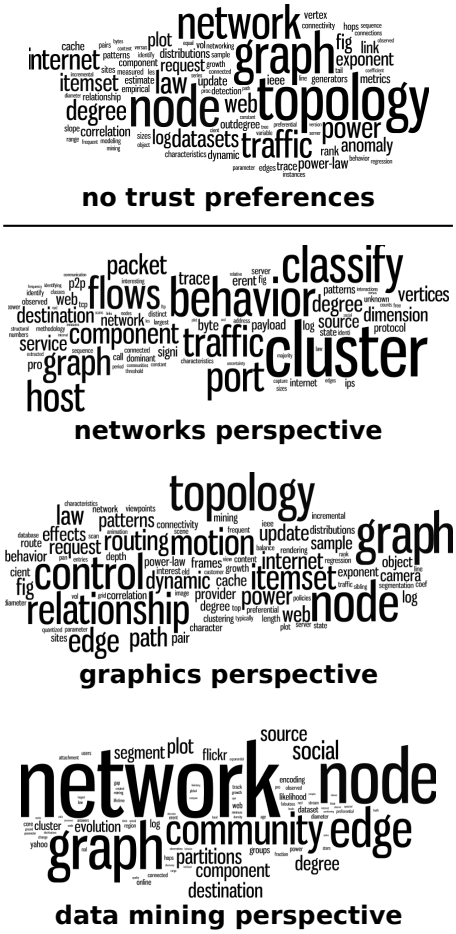


Figure 4.7: A visualization of related work for Faloutsos, Faloutsos, and Faloutsos' "On power-law relationships of the Internet topology." The top word cloud represents papers selected using Equation 4.3, with no trust preferences. (The size of each word in the cloud is proportional to its prevalence in the selected documents.) The subsequent three word clouds represent papers selected using the notion of personalized coverage defined in Equation 4.4, with three different trusted sets $\mathcal{B}$, one representing each author's perspective. Each word cloud visualizes the selected papers that are unique to each author's result set. For example, the bottom word cloud shows the papers found in Christos' data mining-focused results, but do not exist in Petros' or Michalis' result sets.

to answer questions on the usefulness, novelty and trustworthiness of each paper with respect to their research.[9] Additionally, participants were presented with entire result sets and asked to evaluate them in terms of diversity. Figure 4.8 shows the results of the study, from which we can glean the following main observations:

1. On average, users find the papers our algorithm selects to be more useful than those selected by the comparison techniques. The topic modeling approach performs especially poorly, with fewer than half of selected papers deemed useful.

2. Explicitly modeling the individual trust preferences of users leads to more trustworthy papers being selected. However, this comes at the expense of novelty in the selected articles, as researchers are more familiar with the work of authors they trust.

[9]Specific questions asked can be found in Appendix 4.9.

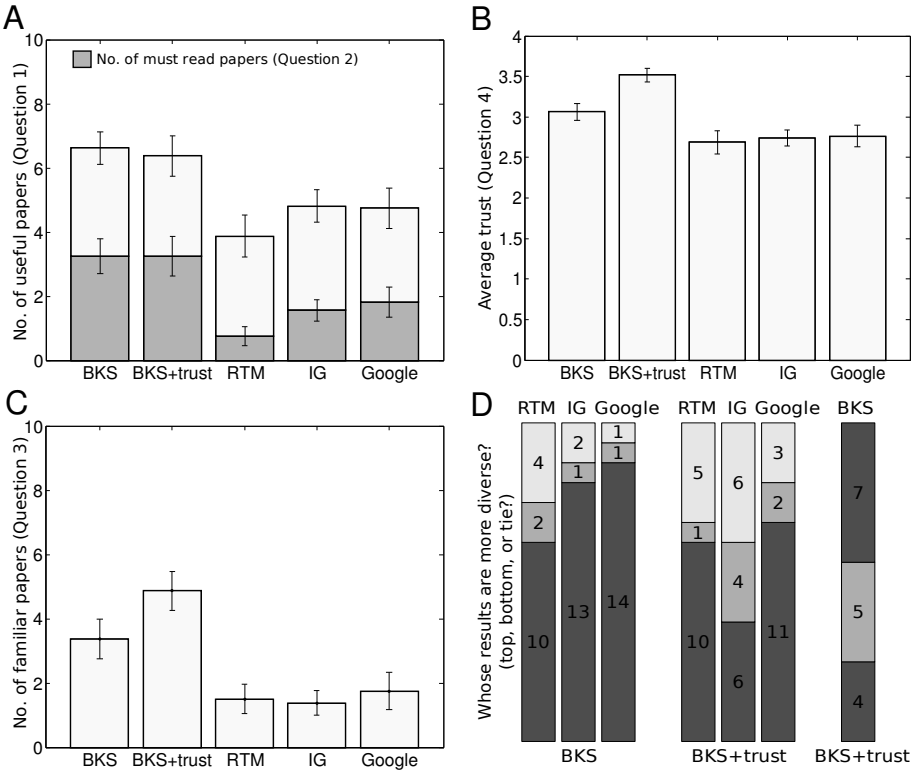


Figure 4.8: User study results comparing two variants of our algorithm, Beyond Keyword Search (BKS), with and without incorporating trust preferences, with the Relational Topic Model (RTM), Information Genealogy (IG) and Google Scholar. Values in bar plots (A), (B) and (C) are responses to the indicated study questions averaged over all sixteen participants, with error bars indicating one standard error. (D) shows how many participants (out of 16) found that our method produced more diverse results compared to the alternative techniques.

3. Our algorithm provides more diverse results than the comparison techniques, which is unsurprising, as our objective functions penalize redundancy.

4.6 Related Work

Researchers in both the library science and computer science communities have studied the shortcomings of the traditional keyword search paradigm [Bates, 1989, Olston and Chi, 2003, Pandit and Olston, 2007]. In fact, our specific query model of defining a researcher's information need as a set of papers rather than as a keyword string has been described before [Bollacker et al., 2000, McNee et al., 2002]. In one particularly related line of research, collaborative filtering techniques that have been successful for movie and product recommendations were adapted to the paper recommendation setting [McNee et al., 2002, Torres et al., 2004]. Another approach uses hypothesis testing to determine the articles that most influence each paper–the paper's *Information Genealogy*–based only on article text [Shaparenko and Joachims, 2007]. Unlike these previous approaches, our methodology is based on a *unified* model of text and citations that places special emphasis on the different trust preferences of individual researchers.

Previous work has also considered the more general, yet related, problem of taking positive examples of membership in a set and using them to expand the set [Elkan and Noto, 2008, Ghahramani and Heller,

2006]. While such approaches have been applied to the domain of research literature, they do not explicitly model the particular characteristics of our problem, e.g., the effect of citations, publication venues and authorship.

Moreover, it is also important to note that our algorithm is *operational* in that it describes a method for selecting papers, in contrast with many *descriptive* studies in bibliometrics, sociology and other fields [Barabási, 2002, de Solla Price, 1965, Newman, 2001a,b, Redner, 1998, Rosvall and Bergstrom, 2008]. In particular, the large body of work on *topic modeling* in computer science and statistics focuses on fitting probabilistic models to document collections by modeling latent themes in the data [Blei and Lafferty, 2009]. While often applied to corpora of scholarly literature [Airoldi et al., 2010, Blei and Lafferty, 2006, 2007, Dietz et al., 2007, Erosheva et al., 2004, Gerrish and Blei, 2010, Griffiths and Steyvers, 2004, Rozen-Zvi et al., 2004], paper recommendation is not the primary objective of these models. Rather, our algorithm follows from a line of work that frames document selection as an explicit optimization problem (cf. [El-Arini et al., 2009]).

Finally, we note that the approach we describe in this chapter is, in fact, agnostic to the specific definition of influence we use, and thus while we define influence to have an explicit probabilistic interpretation, other such definitions are possible. For instance, recent work by Lao and Cohen [2010] provides an approach based on path-constrained random walks, which we can plug in as an alternative definition for influence.

4.7 Conclusions

The problem addressed in this chapter—how to move beyond keyword search when trying to find relevant scientific literature— very much inspired the title of this thesis. Here, we have a critical information retrieval task that is query-dependent, but the most natural form of querying does not correspond to the traditional keyword queries we are used to when it comes to Web search.

We hypothesized that the interactive concept coverage methodology that we introduced in the previous chapters was flexible enough to successfully address this qualitatively different problem. This was evident in that the underlying objective function and probabilistic set coverage functions remained mostly the same between this chapter and the last, resulting in identical optimization algorithms and theoretical guarantees. However, in this problem setting, we had the opportunity to examine two different aspects of our six-step methodology that we had not encountered before:

- While the news recommendation setting is queryless, here we showed how to successfully incorporate a complex query structure into our approach, defining an *augmented concept representation* and corresponding document coverage function that captured a domain-specific idea of *influence*.
- Personal trust preferences play an important role when scholars decide which articles to read or cite. However, much of these preferences and affinities are based on side information, not on the actual content that we are covering with our objective function. As such, we showed how we can define a probabilistic *personalized affinity function*, which we seamlessly incorporate into our notion of document coverage, that incentivized our objective function to select documents that were more likely to be trusted by the user.

In both of these cases, we moved beyond the simple user interaction process of the previous chapter, instead soliciting richer information from users in non-onerous, natural methods.

The success of our interactive concept coverage methodology at addressing this problem was demonstrated by our experimental results, where an in-depth user study showed that the articles recommended by our approach were more relevant, more trustworthy and more diverse than those recommended by a variety of state-of-the-art approaches, including the current gold standard of Google Scholar. We envision this particular application of our methodology to become an important complement to keyword search for the scholarly research community.

4.8 Appendix: Data Details and Preprocessing

In this chapter we refer to two data sets of scientific publications:

1. *PNAS Data*: Five years worth of articles from the Proceedings of the National Academy of Sciences (1997-2001; 13,648 papers).

2. *ACM Data*: A subset of the Association for Computing Machinery Digital Library, focused on papers in machine learning and related areas (1959-2009; 35,042 papers).

Both data sets include the title, authors, publication date, venue or publication track, citations, abstract and full text (when available) of each paper. The particular running example in this chapter refers to the PNAS articles in Table 4.1.

Each data set was preprocessed to ensure the acyclicity of the citation graph, as well as to extract a vocabulary $\mathcal{C}$ of important concepts. The content of each paper is represented as a frequency vector of these selected concepts.

Processing the citation graph. Based on simple chronology, one would expect a citation graph to be acyclic; after all, a researcher cannot cite a paper if it does not yet exist. However, this is not quite the case in practice. For instance, colleagues writing several papers simultaneously may cite each other, leading to doubly-connected pairs in the graph. As our algorithms rely on the acyclicity of the citation graph, we take the following steps to remove cycles:

1. Remove self cycles from the graph (i.e., edges that start and end at the same node).

2. Find the strongly connected components (SCCs) of the graph (i.e., maximal subgraphs such that for any two nodes x, y in the subgraph, there is a path from x to y and a path from y to x). In a directed acyclic graph (DAG), all the SCCs are of size one. However, this is generally not the case in real citation graphs.

3. For SCCs of size two (i.e., "I cite you and you cite me"), we employ the following heuristic to determine which edge to cut:

    - If the two papers were published in different years, have the later paper cite the earlier paper.

    - Else, if number of citations is different, have the lesser cited paper cite the more highly cited paper.

    - Else, pick one of the two edges uniformly at random.

4. While the previous step takes care of most cycles, a few peculiar cases with SCCs of size greater than two usually remain. There are few enough of these that we look at each such component individually, and manually decide which edges to cut.

Finally, recall that we augment the citation graph with edges indicating common authorship. In this step, we only connect papers that were written within five years of each other, as influence may tend to diminish over time. Moreover, when augmenting the graph with these edges, we ensure that we are not creating any cycles.

Selecting concepts. A typical corpus of scientific publications may contain tens of thousands of unique words, but only a fraction of them will be informative. Thus, working with the entire set of words rather than a particular subset can be wasteful. To this end, for each data set, we select a subset of words that we use as *concepts*:

- Ignore stop words (e.g., "the," "and," "of," etc.), words containing non-alphanumeric characters, and words that are too long (> 20 characters) or too short (< 3 characters).
- Of the remaining words, select the top 10,000 most frequent.
- Of these words, select ones that appear in at least 40 articles but fewer than 3,500 articles. If a word appears in too few articles, it is likely to be overly specific, while if it appears in a large fraction of articles, it is likely to be too general (e.g., the word "cell" for the case of PNAS, or "computer" for ACM). (These numbers are for the PNAS data set. For the larger ACM collection, we require words to appear in at least 100 documents and in no more than 8,000.)
- Finally, in an attempt to avoid selecting marginal words, we only select words such that when they appear in a document, they appear at least twice (on average).

4.9 Appendix: User Study Details

The following questions were asked of each user study participant, for each article presented:

1. Assume you came across this paper while working on the study paper. From reading the title and abstract, would you have been inclined to:
   (a) continue reading the paper (even if just to skim), because you think it might be useful to the work of the study paper?
   (b) walk away (i.e., from the title and abstract alone, you can already tell that this paper is not useful to the work of the study paper)?
2. Do you feel that this paper would have been a *must read* for you when working on the study paper? (i.e., you would have read this paper carefully had you known about it, and perhaps would have cited it) [Yes, No]
3. Did you know about this paper before? [Yes, No]
4. Taking the authors and venue into account, would you be inclined to trust what this paper has to say?
   (a) For sure [4]

(b) Probably [3]

(c) Not sure [2]

(d) Probably not [1]

(e) Not at all [0]

(Figure 4.8A plots the responses to questions 1 and 2. Figure 4.8B plots the responses to question 4. Figure 4.8C plots the responses to question 3.)

After answering these questions for all papers selected by all five approaches, the participant is presented with all ten pairings of the five approaches, head to head, one pair of result sets at a time (e.g., RTM results on the left of the screen, our results with trust on the right). For each pair of result sets, the participant is asked to indicate which of the result sets is more diverse, or if they are equally diverse. As a diverse set of useless results is not beneficial to a researcher, in this part of the study we only display the papers that were indicated as useful by the participant in the previous section (i.e., an affirmative answer to question 1). (These diversity results are plotted in Figure 4.8D.)

4.10 Appendix: Selected Papers

Tables 4.2-4.4 show the papers selected for our running PNAS example. In particular, Tables 4.3 and 4.4 show the papers presented in Figure 4.6. Table 4.5 provides the papers selected for the example in Figure 4.1D. Tables 4.6-4.9 show the papers selected for the example in Figure 4.7.

Table 4.1: Articles from PNAS example

ID	Title	Year	Volume	Pages
160	Physiological reactions of nitric oxide and hemoglobin: A radical rethink	1999	96	9967-9969
244	Isolation of a temperate bacteriophage encoding the type III effector protein SopE from an epidemic *Salmonella typhimurium* strain	1999	96	9845-9850
292	Bacteriophages in the evolution of pathogen-host interactions	1999	96	9452-9454
1139	Nitroreductase A is regulated as a member of the *soxRS* regulon of *Escherichia coli*	1999	96	3537-3539
1304	A mechanism of paraquat toxicity involving nitric oxide synthase	1999	96	12760-12765
1839	Ancient origins of nitric oxide signaling in biological systems	1999	96	14206-14207
2094	Hemoglobin induction in mouse macrophages	1999	96	6643-6647
2136	Virulent *Salmonella typhimurium* has two periplasmic Cu, Zn-superoxide dismutases	1999	96	7502-7507
2389	A highly conserved sequence is a novel gene involved in *de novo* vitamin B6 biosynthesis	1999	96	9374-9378
2452	The oxyhemoglobin reaction of nitric oxide	1999	96	9027-9032
4468	Periplasmic superoxide dismutase protects *Salmonella* from products of phagocyte NADPH-oxidase and nitric oxide synthase	1997	94	13997-14001
5550	Nitric oxide in plant immunity	1998	95	10345-10347
5688	Defense gene induction in tobacco by nitric oxide, cyclic GMP, and cyclic ADP-ribose	1998	95	10328-10333
7273	Roles for mannitol and mannitol dehydrogenase in active oxygen-mediated plant defense	1998	95	15129-15133
8305	*S*-nitrosothiol repletion by an inhaled gas regulates pulmonary function	2001	98	5792-5797
8365	Flavohemoglobin denitrosylase catalyzes the reaction of a nitroxyl equivalent with molecular oxygen	2001	98	10108-10112
8445	Expression and phylogeny of claudins in vertebrate primordia	2001	98	10196-10201
8490	Peptide methionine sulfoxide reductase from *Escherichia coli* and *Mycobacterium tuberculosis* protects bacteria against oxidative damage from reactive nitrogen intermediates	2001	98	9901-9906

Table 4.1 (cont.): Articles from PNAS example

ID	Title	Year	Volume	Pages
8643	Plant mitogen-activated protein kinase cascades: Negative regulatory roles turn out positive	2001	98	784-786
8853	Myoglobin: A scavenger of bioactive NO	2001	98	735-740
8901	Simultaneous observation of the O—O and Fe—O_2 stretching modes in oxyhemoglobins	2001	98	479-484
8910	Activation of a mitogen-activated protein kinase pathway is involved in disease resistance in tobacco	2001	98	741-746
9135	Catalytic consumption of nitric oxide by 12/15- lipoxygenase: Inhibition of monocyte soluble guanylate cyclase activation	2001	98	8006-8011
9318	*Helicobacter pylori* arginase inhibits nitric oxide production by eukaryotic cells: A strategy for bacterial survival	2001	98	13844-13849
9429	Reciprocal electromechanical properties of rat prestin: The motor molecule from rat outer hair cells	2001	98	4178-4183
9452	B lymphocyte-restricted expression of prion protein does not enable prion replication in prion protein knockout mice	2001	98	4034-4037
9467	Plasma nitrite rather than nitrate reflects regional endothelial nitric oxide synthase activity but lacks intrinsic vasodilator action	2001	98	12814-12819
9573	Supermolecular structure of the enteropathogenic *Escherichia coli* type III secretion system and its direct interaction with the EspA-sheath-like structure	2001	98	11638-11643
9582	Modulation of nitric oxide bioavailability by erythrocytes	2001	98	11771-11776
9625	Cysteine-3635 is responsible for skeletal muscle ryanodine receptor modulation by NO	2001	98	11158-11162
9890	*In vivo* mechanism-based inactivation of *S*-adenosylmethionine decarboxylases from *Escherichia coli*, *Salmonella typhimurium*, and *Saccharomyces cerevisiae*	2001	98	10578-10583
10008	Structure of sortase, the transpeptidase that anchors proteins to the cell wall of *Staphylococcus aureus*	2001	98	6056-6061
10090	Comparison of a hair bundle's spontaneous oscillations with its response to mechanical stimulation reveals the underlying active process	2001	98	14380-14385
10118	Compressive nonlinearity in the hair bundle's active response to mechanical stimulation	2001	98	14386-14391
10123	*In vivo* evidence for a cochlear amplifier in the hair-cell bundle of lizards	2001	98	2826-2831

Table 4.1 (cont.): Articles from PNAS example

ID	Title	Year	Volume	Pages
10161	Defective localization of the NADPH phagocyte oxidase to *Salmonella*-containing phagosomes in tumor necrosis factor p55 receptor-deficient macrophages	2001	98	2561-2565
10372	Regulation of the *Mycobacterium tuberculosis* hypoxic response gene encoding α-crystallin	2001	98	7534-7539
10605	Physical basis of two-tone interference in hearing	2001	98	9080-9085
10642	A fatty acid desaturase modulates the activation of defense signaling pathways in plants	2001	98	9448-9453
10693	Scrapie prion protein accumulation by scrapie-infected neuroblastoma cells abrogated by exposure to a prion protein antibody	2001	98	9295-9299
10844	Neuroglobin is up-regulated by and protects neurons from hypoxic-ischemic injury	2001	98	15306-15311
10850	Oxygen radical inhibition of nitric oxide-dependent vascular function in sickle cell disease	2001	98	15215-15220
10900	Epitope tagging of chromosomal genes in *Salmonella*	2001	98	15264-15269
10940	Polymerization of a single protein of the pathogen *Yersinia enterocolitica* into needles punctures eukaryotic cells	2001	98	4669-4674
11134	Relative role of heme nitrosylation and β-cysteine 93 nitrosation in the transport and metabolism of nitric oxide by hemoglobin in the human circulation	2000	97	9943-9948
11770	Protection from nitrosative stress by yeast flavohemoglobin	2000	97	4672-4676
11791	The *Pseudomonas syringae* Hrp pathogenicity island has a tripartite mosaic structure composed of a cluster of type III secretion genes bounded by exchangeable effector and conserved effector loci that contribute to parasitic fitness and pathogenicity in plants	2000	97	4856-4861
12134	*Arabidopsis* RelA/SpoT homologs implicate (p)ppGpp in plant signaling	2000	97	3747-3752
12176	Cochlear mechanisms from a phylogenetic viewpoint	2000	97	11736-11743
12270	Putting ion channels to work: Mechanoelectrical transduction, adaptation, and amplification by hair cells	2000	97	11765-11772
12286	Molecular mechanisms of sound amplification in the mammalian cochlea	2000	97	11759-11764
12379	Contribution of *Salmonella typhimurium* type III secretion components to needle complex formation	2000	97	11008-11013
13042	A conserved amino acid sequence directing intracellular type III secretion by *Salmonella typhimurium*	2000	97	7539-7544

Table 4.1 (cont.): Articles from PNAS example

ID	Title	Year	Volume	Pages
13204	Reactive oxygen and nitrogen intermediates in the relationship between mammalian hosts and microbial pathogens	2000	97	8841-8848
13240	The *Arabidopsis dnd1* "defense, no death" gene encodes a mutated cyclic nucleotide-gated ion channel	2000	97	9323-9328
13264	Nitric oxide and salicylic acid signaling in plant defense	2000	97	8849-8855
13279	Genetic complexity of pathogen perception by plants: The example of *Rcr3*, a tomato gene required specifically by *Cf-2*	2000	97	8807-8814
13283	*Pseudomonas syringae* Hrp type III secretion system and effector proteins	2000	97	8770-8777
13316	Nitric oxide prevents cardiovascular disease and determines survival in polyglobulic mice overexpressing erythropoietin	2000	97	11609-11613
13344	Role of circulating nitrite and *S*-nitrosohemoglobin in the regulation of regional blood flow in humans	2000	97	11482-11487

Table 4.2: Selected papers for PNAS example (no trust)

Rank	Title	Year	Volume	Pages
1	Nitric oxide in plant immunity	1998	95	10345-10347
2	Defective localization of the NADPH phagocyte oxidase to *Salmonella*-containing phagosomes in tumor necrosis factor p55 receptor-deficient macrophages	2001	98	2561-2565
3	Ancient origins of nitric oxide signaling in biological systems	1999	96	14206-14207
4	Virulent *Salmonella typhimurium* has two periplasmic Cu, Zn-superoxide dismutases	1999	96	7502-7507
5	Nitric oxide and salicylic acid signaling in plant defense	2000	97	8849-8855
6	Reactive oxygen and nitrogen intermediates in the relationship between mammalian hosts and microbial pathogens	2000	97	8841-8848
7	A mechanism of paraquat toxicity involving nitric oxide synthase	1999	96	12760-12765
8	Roles for mannitol and mannitol dehydrogenase in active oxygen-mediated plant defense	1998	95	15129-15133
9	The *Arabidopsis dnd1* "defense, no death" gene encodes a mutated cyclic nucleotide-gated ion channel	2000	97	9323-9328
10	*Arabidopsis* RelA/SpoT homologs implicate (p)ppGpp in plant signaling	2000	97	3747-3752

Table 4.3: Selected papers for PNAS example (as a plant biologist)

Rank	Title	Year	Volume	Pages
1	Nitric oxide and salicylic acid signaling in plant defense	2000	97	8849-8855
2	Ancient origins of nitric oxide signaling in biological systems	1999	96	14206-14207
3	The *Arabidopsis dnd1* "defense, no death" gene encodes a mutated cyclic nucleotide-gated ion channel	2000	97	9323-9328
4	Roles for mannitol and mannitol dehydrogenase in active oxygen-mediated plant defense	1998	95	15129-15133
5	Defective localization of the NADPH phagocyte oxidase to *Salmonella*-containing phagosomes in tumor necrosis factor p55 receptor-deficient macrophages	2001	98	2561-2565
6	*Arabidopsis* RelA/SpoT homologs implicate (p)ppGpp in plant signaling	2000	97	3747-3752
7	A fatty acid desaturase modulates the activation of defense signaling pathways in plants	2001	98	9448-9453
8	Reactive oxygen and nitrogen intermediates in the relationship between mammalian hosts and microbial pathogens	2000	97	8841-8848
9	Virulent *Salmonella typhimurium* has two periplasmic Cu, Zn-superoxide dismutases	1999	96	7502-7507
10	A highly conserved sequence is a novel gene involved in *de novo* vitamin B6 biosynthesis	1999	96	9374-9378

Table 4.4: Selected papers for PNAS example (as an immunologist)

Rank	Title	Year	Volume	Pages
1	Defective localization of the NADPH phagocyte oxidase to *Salmonella*-containing phagosomes in tumor necrosis factor p55 receptor-deficient macrophages	2001	98	2561-2565
2	Virulent *Salmonella typhimurium* has two periplasmic Cu, Zn-superoxide dismutases	1999	96	7502-7507
3	Reactive oxygen and nitrogen intermediates in the relationship between mammalian hosts and microbial pathogens	2000	97	8841-8848
4	*Helicobacter pylori* arginase inhibits nitric oxide production by eukaryotic cells: A strategy for bacterial survival	2001	98	13844-13849
5	Nitric oxide and salicylic acid signaling in plant defense	2000	97	8849-8855
6	Nitric oxide in plant immunity	1998	95	10345-10347
7	Peptide methionine sulfoxide reductase from *Escherichia coli* and *Mycobacterium tuberculosis* protects bacteria against oxidative damage from reactive nitrogen intermediates	2001	98	9901-9906
8	Ancient origins of nitric oxide signaling in biological systems	1999	96	14206-14207
9	The oxyhemoglobin reaction of nitric oxide	1999	96	9027-9032
10	A mechanism of paraquat toxicity involving nitric oxide synthase	1999	96	12760-12765

Table 4.5: Selected papers for example in Figure 4.1D

Rank	Title	Year	Volume	Pages
1	Defense gene induction in tobacco by nitric oxide, cyclic GMP, and cyclic ADP-ribose	1998	95	10328-10333
2	Ancient origins of nitric oxide signaling in biological systems	1999	96	14206-14207
3	Periplasmic superoxide dismutase protects *Salmonella* from products of phagocyte NADPH-oxidase and nitric oxide synthase	1997	94	13997-14001
4	A mechanism of paraquat toxicity involving nitric oxide synthase	1999	96	12760-12765
5	Nitroreductase A is regulated as a member of the *soxRS* regulon of *Escherichia coli*	1999	96	3537-3539
6	Nitric oxide and salicylic acid signaling in plant defense	2000	97	8849-8855
7	*S*-nitrosothiol repletion by an inhaled gas regulates pulmonary function	2001	98	5792-5797
8	Cysteine-3635 is responsible for skeletal muscle ryanodine receptor modulation by NO	2001	98	11158-11162
9	The oxyhemoglobin reaction of nitric oxide	1999	96	9027-9032
10	Protection from nitrosative stress by yeast flavohemoglobin	2000	97	4672-4676
11	Hemoglobin induction in mouse macrophages	1999	96	6643-6647
12	Physiological reactions of nitric oxide and hemoglobin: A radical rethink	1999	96	9967-9969
13	Cochlear mechanisms from a phylogenetic viewpoint	2000	97	11736-11743
14	Plant mitogen-activated protein kinase cascades: Negative regulatory roles turn out positive	2001	98	784-786
15	Flavohemoglobin denitrosylase catalyzes the reaction of a nitroxyl equivalent with molecular oxygen	2001	98	10108-10112
16	Relative role of heme nitrosylation and β-cysteine 93 nitrosation in the transport and metabolism of nitric oxide by hemoglobin in the human circulation	2000	97	9943-9948
17	Role of circulating nitrite and *S*-nitrosohemoglobin in the regulation of regional blood flow in humans	2000	97	11482-11487
18	Modulation of nitric oxide bioavailability by erythrocytes	2001	98	11771-11776
19	Nitric oxide prevents cardiovascular disease and determines survival in polyglobulic mice overexpressing erythropoietin	2000	97	11609-11613
20	Plasma nitrite rather than nitrate reflects regional endothelial nitric oxide synthase activity but lacks intrinsic vasodilator action	2001	98	12814-12819

Table 4.6: Selected papers for example in Figure 4.7 (unpersonalized)

Rank	Title	Authors	Year
1	Prediction of future world wide web traffic characteristics for capacity planning	Christensen, Javagal	1997
2	Self-similarity in World Wide Web traffic: evidence and possible causes	Crovella, Bestavros	1997
3	Characteristics of WWW Client-based Traces	Cunha et al.	1995
4	Empirically derived analytic models of wide-area TCP connections	Paxson	1994
5	End-to-end available bandwidth as a random autocorrelated QoS-relevant time-series	Chobanyan et al.	2008
6	Efficiently serving dynamic data at highly accessed web sites	Challenger et al.	2004
7	A Prefetching Protocol Using Client Speculation for the WWW	Bestavros, Cunha	1995
8	Power laws and the AS-level internet topology	Siganos et al.	2003
9	Power-law relationship and self-similarity in the itemset support distribution: analysis and applications	Chuang et al.	2008
10	On the origin of power laws in Internet topologies	Medina et al.	2000
11	Spatio-temporal network anomaly detection by assessing deviations of empirical measures	Paschalidis, Smaragdakis	2009
12	Network topology generators: degree-based vs. structural	Tangmunarunkit et al.	2002
13	Mathematical models for academic webs: linear relationship or non-linear power law?	Payne, Thelwall	2005
14	A random graph model for massive graphs	Aiello et al.	2000

Table 4.7: Selected papers for example in Figure 4.7 (networks)

Rank	Title	Authors	Year
1	Self-similarity in World Wide Web traffic: evidence and possible causes	Crovella, Bestavros	1997
2	Empirically derived analytic models of wide-area TCP connections	Paxson	1994
3	Power laws and the AS-level internet topology	Siganos et al.	2003
4	Characteristics of WWW Client-based Traces	Cunha et al.	1995
5	Weighted graphs and disconnected components: patterns and a generator	McGlohon et al.	2008
6	Learning for accurate classification of real-time traffic	Li, Moore	2006
7	BLINC: multilevel traffic classification in the dark	Karagiannis et al.	2005
8	Graphs over time: densification laws, shrinking diameters and possible explanations	Leskovec et al.	2005
9	Graph evolution: Densification and shrinking diameters	Leskovec et al.	2007
10	A Prefetching Protocol Using Client Speculation for the WWW	Bestavros, Cunha	1995
11	Scalable modeling of real graphs using Kronecker multiplication	Leskovec, Faloutsos	2007
12	ANF: a fast and scalable tool for data mining in massive graphs	Palmer et al.	2002
13	A random graph model for massive graphs	Aiello et al.	2000
14	Profiling internet backbone traffic: behavior models and applications	Xu et al.	2005

Table 4.8: Selected papers for example in Figure 4.7 (graphics)

Rank	Title	Authors	Year
1	Characteristics of WWW Client-based Traces	Cunha et al.	1995
2	Power laws and the AS-level internet topology	Siganos et al.	2003
3	Empirically derived analytic models of wide-area TCP connections	Paxson	1994
4	ANF: a fast and scalable tool for data mining in massive graphs	Palmer et al.	2002
5	Self-similarity in World Wide Web traffic: evidence and possible causes	Crovella, Bestavros	1997
6	Weighted graphs and disconnected components: patterns and a generator	McGlohon et al.	2008
7	Parallax photography: creating 3D cinematic effects from stills	Zheng et al.	2009
8	On inferring autonomous system relationships in the internet	Gao	2001
9	Power-law relationship and self-similarity in the itemset support distribution: analysis and applications	Chuang et al.	2008
10	On the origin of power laws in Internet topologies	Medina et al.	2000
11	Composable controllers for physics-based character animation	Faloutsos et al.	2001
12	Segmenting motion capture data into distinct behaviors	Barbič et al.	2004
13	Efficiently serving dynamic data at highly accessed web sites	Challenger et al.	2004
14	Graph mining: Laws, generators, and algorithms	Chakrabarti, Faloutsos	2006

Table 4.9: Selected papers for example in Figure 4.7 (data mining)

Rank	Title	Authors	Year
1	Characteristics of WWW Client-based Traces	Cunha et al.	1995
2	Power laws and the AS-level internet topology	Siganos et al.	2003
3	Graph evolution: Densification and shrinking diameters	Leskovec et al.	2007
4	Graphs over time: densification laws, shrinking diameters and possible explanations	Leskovec et al.	2005
5	Weighted graphs and disconnected components: patterns and a generator	McGlohon et al.	2008
6	Self-similarity in World Wide Web traffic: evidence and possible causes	Crovella, Bestavros	1997
7	Microscopic evolution of social networks	Leskovec et al.	2008
8	Statistical properties of community structure in large social and information networks	Leskovec et al.	2008
9	Scalable modeling of real graphs using Kronecker multiplication	Leskovec, Faloutsos	2007
10	Empirically derived analytic models of wide-area TCP connections	Paxson	1994
11	ANF: a fast and scalable tool for data mining in massive graphs	Palmer et al.	2002
12	Structure and evolution of online social networks	Kumar et al.	2006
13	Visualization of large networks with min-cut plots, A-plots and R-MAT	Chakrabarti et al.	2007
14	GraphScope: parameter-free mining of large time-evolving graphs	Sun et al.	2007

Chapter 5

Transparent User Models for Personalization

In Chapter 1 of this thesis, we argued that, while personalization might be necessary in today's world of information overload, it comes with a litany of problems that must be addressed in order to achieve better results. Many of these problems were technical in nature, from cold starts to redundancy to complex queries. Chapters 3 and 4 focused on how we could use our interactive concept coverage methodology to address these technical concerns, and we showed that we could do so successfully.

However, in addition to the technical problems of personalization, we described social issues with such technology that remain unsettled. As mentioned in the introduction to this thesis, the Pew Internet and American Life Project reports that many Americans have an unfavorable view of personalization, finding it to be an invasion of privacy.[1] Additionally:

- Users often do not know that their results are being personalized in the first place, and as such, may not understand why their Web experience is different from (and perhaps worse than) that of their friends.
- Even if they are aware that their results are personalized, users are rarely provided with information about how the particular site or service perceives them, and, as such, have little recourse to make corrections, if necessary. For example, in a famous critique of personalization as applied to television show recommendation in TiVo,[2] Zaslow describes the drastic steps users feel they need to take in order to correct misperceptions that the system has of them, reaching the conclusion that "there's just one way to change its 'mind': outfox it," [Zaslow, 2002].
- Even when a system correctly models a user's interests and tastes, it may not always be desirable to use such information. Whether because of privacy concerns or to avoid groupthink (cf. Pariser's *The Filter Bubble* [2011]), users may wish to selectively inhibit certain signals or attributes from being used for personalization.

In this chapter, we seek to address these social challenges by making personalization more *transparent*. In other words, users should know, (1) *when* personalization is happening, and, (2) *how* they are perceived by

[1] http://www.pewinternet.org/Reports/2012/Search-Engine-Use-2012/Summary-of-findings.aspx

[2] http://www.tivo.com

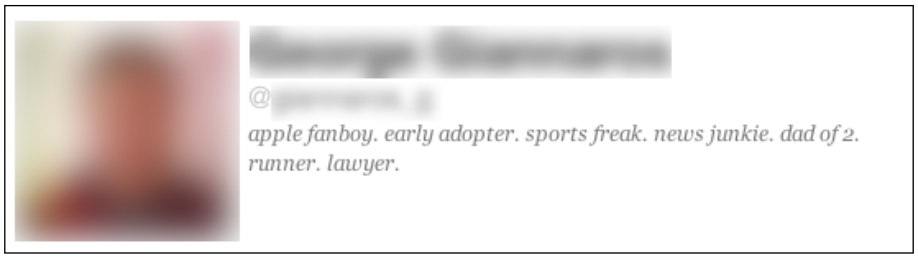


Figure 5.1: An example Twitter profile, showing an anonymized user's self-reported description. Here, if we are interested in the "Apple fanboy" badge, we can observe this user's actions on Twitter to help us figure out what it means to be an Apple fanboy.

the system (with the ability to correct this perception as necessary).

We provide this transparency by representing each user as a set of interpretable, explainable attributes (e.g., "vegetarian," "hipster," or "Apple fanboy") that we learn from user behavior. It is important that both the *meaning* of these attributes and *why they were assigned* to the user be readily apparent. Following the paradigm made popular by location-aware social networks such as Foursquare,[3] we refer to these attributes as *badges*.

In particular, we consider the microblogging site Twitter,[4] and we associate each badge with a characteristic *label* (e.g., "Apple fanboy") that a user might use to describe himself in his Twitter profile. For any user, we can observe his or her profile and determine whether or not it contains a particular label. For example, the user profile in Figure 5.1 contains the label "Apple fanboy," which we might associate with an Apple fanboy badge. It is important to note that there is a *probabilistic* relationship between labels and the badges they correspond to. For example, most users who adore Apple products will not explicitly identify themselves with "Apple fanboy" in their Twitter profiles.[5] Nevertheless, we wish to use the actions of those *self-identified* Apple fanboys to help us learn what it means to be one. We can then hope to predict which other users might also be Apple fanboys, even if they don't identify themselves as such.

Moreover, in this chapter, we take the view that the set of possible badges (and their corresponding labels) are defined *a priori* in a supervised manner. Specifically, we assume we are *given* some set of possible badges (e.g., as in Table 5.1), and wish to *infer*: (1) their presence or absence for each user, and, (2) how they manifest themselves in terms of Twitter behavior.

In the remainder of this chapter, we describe how we learn badges from user activity on Twitter, using a Bayesian framework to explicitly model uncertainty. We show experimental results on real Twitter users, and present both quantitative evidence and qualitative anecdotes demonstrating the effectiveness of our method at producing transparent user models. Next, in Chapter 6, we will build upon the ideas developed here to define a novel concept representation that takes advantage of such transparency.

[3] `http://www.foursquare.com`

[4] `http://www.twitter.com`

[5] This stands in contrast to Foursquare, where badges are deterministic (e.g., five check-ins at an airport *guarantee* the "jet-setter" badge).

5.1 Modeling Badges

We describe each user as a set of *latent* badges that, collectively, explain the user's behavior. The fundamental problem we seek to solve is: *how do we infer the badges for each user based on observed actions and labels?*

For each user u, we observe two binary vectors:

1. The label vector $\boldsymbol{\lambda}^{(u)}$, with $\lambda_i^{(u)} = 1$ indicating that the Twitter profile of user u contains the label corresponding to badge i. As an example, if we let badge i correspond to the "runner" label, then $\lambda_i^{(u)} = 1$ if the Twitter profile of user u contains the word "runner," and $\lambda_i^{(u)} = 0$ otherwise.

2. The action vector $\boldsymbol{a}^{(u)}$, where $a_j^{(u)} = 1$ if user u is observed performing action j. In our Twitter domain, we take the set of possible actions to include all hashtags and retweets. For example, action j might correspond to tweeting with the hashtag #`runkeeper`, and $a_j^{(u)} = 1$ for users u that have such a tweet.

We model these observations as probabilistically arising from a latent set of badges $\boldsymbol{b}^{(u)}$, where $b_i^{(u)} \in \{0, 1\}$ indicates whether or not user u has badge i. In particular, we elect to define a generative—rather than discriminative—model; while the high precision labels may provide us with positive training examples, their low recall leads to no meaningful negative examples. Moreover, if a user chooses to decline a badge that we predict for him (e.g., he might not really be an Apple fanboy), this simply corresponds in our model to observing the latent variable $b_i^{(u)} = 0$. Additionally, we note that our model differs from traditional unsupervised latent variable models, such as topic models [Blei and Lafferty, 2009], in that the badge labels provide identifiability that we would not otherwise achieve. Thus, for example, if we define the label for badge i to correspond to those users with "runner" in their Twitter profile, then the actions explained by badge i will always correspond to (our view of) runners, which is a property we do not get with fully unsupervised topic models, such as latent Dirichlet allocation [Blei et al., 2003].

5.1.1 Generating labels

Given a particular user's badge assignments, the generative process for labels encodes our intuition that each label $\boldsymbol{\lambda}_i$ is a high precision, low recall indicator of the presence or absence of a badge. Specifically, "high precision" here means that it is very unlikely for someone without badge i (i.e., with $b_i^{(u)} = 0$) to use the corresponding label (i.e., $\lambda_i^{(u)} = 1$) in his profile, while "low recall" indicates that many users u with $b_i^{(u)} = 1$ nevertheless have $\lambda_i^{(u)} = 0$. For example, while most vegetarians on Twitter do not describe themselves as "vegetarian" in their Twitter profiles, it is much more rare (but not impossible) for non-vegetarians to have the word "vegetarian" in their profiles.

As such, we model label $\lambda_i^{(u)}$ as being *a priori* present with a true positive rate γ_i^{T} and false positive rate γ_i^{F} (with $\gamma_i^{\mathrm{F}} \ll \gamma_i^{\mathrm{T}} < 1$ and $\gamma_i^{\mathrm{F}} \approx 0$). Formally, we have:

$$p(\lambda_i^{(u)} = 1 \mid b_i^{(u)}, \gamma_i^{\mathrm{T}}, \gamma_i^{\mathrm{F}}) = \mathsf{Bernoulli}(b_i^{(u)}\gamma_i^{\mathrm{T}} + (1 - b_i^{(u)})\gamma_i^{\mathrm{F}}),$$

given the user's badge $b_i^{(u)}$. In other words, the presence of a badge does not necessarily imply its appearance in a user's profile, and it is precisely these badges that we aim to infer.

5.1.2 Generating actions

We assume that the observed actions $a_j^{(u)} \in \{0, 1\}$ of a user u can be explained by one or more of his latent badges $b_i^{(u)}$. In the Twitter domain, possible actions j might include a user *re-tweeting* some author, or using a particular *hashtag*.

For each possible badge i and action j, there is a probability $s_{ij}\phi_{ij}$ of associating them; it is decomposed into a context-specific rate $\phi_{ij} \in (0, 1)$ and a sparsity prior $s_{ij} \in \{0, 1\}$. The $\mathbf{s}_i$ variables for a badge i act as a mask, delineating which actions can be explained by this particular badge, and their sparsity is controlled by a badge-specific prior η_i. Given that $s_{ij} = 1$, the variable ϕ_{ij} represents the probability that a user with badge i undertakes action j, in the absence of any other badges. For example, if a user only has the "runner" badge active, and $s_{\text{runner},\#\texttt{runkeeper}} = 1$, meaning that the "runner" badge can explain tweeting `#runkeeper`, then our user will tweet `#runkeeper` with probability $\phi_{\text{runner},\#\texttt{runkeeper}}$.

As a user may have more than one badge active that can explain a particular action, we combine their influence in a noisy-or fashion, indicating that a user performs an action j if at least one of his badges induce him to do so. Moreover, it is plausible that a user's behavior is influenced not just by his particular attributes, but by the environment at large, and thus we assume a background model $\phi_{\text{bg},j}$, acting as a badge that every user shares, that has some probability of explaining every action.

Thus, formally, action $a_j^{(u)}$ is observed if it is explained by either the background or at least one of the badges of user u, which we can write as follows:[6]

$$p(a_j^{(u)} = 1 \mid \boldsymbol{b}^{(u)}, \boldsymbol{\phi}_{\bullet j}, \mathbf{s}_{\bullet j}) = \\ \mathsf{Bernoulli}\left(1 - (1 - \phi_{\text{bg},j}) \prod_{i:b_i^{(u)}=1} (1 - \phi_{ij} s_{ij})\right).$$

5.1.3 Prior probabilities

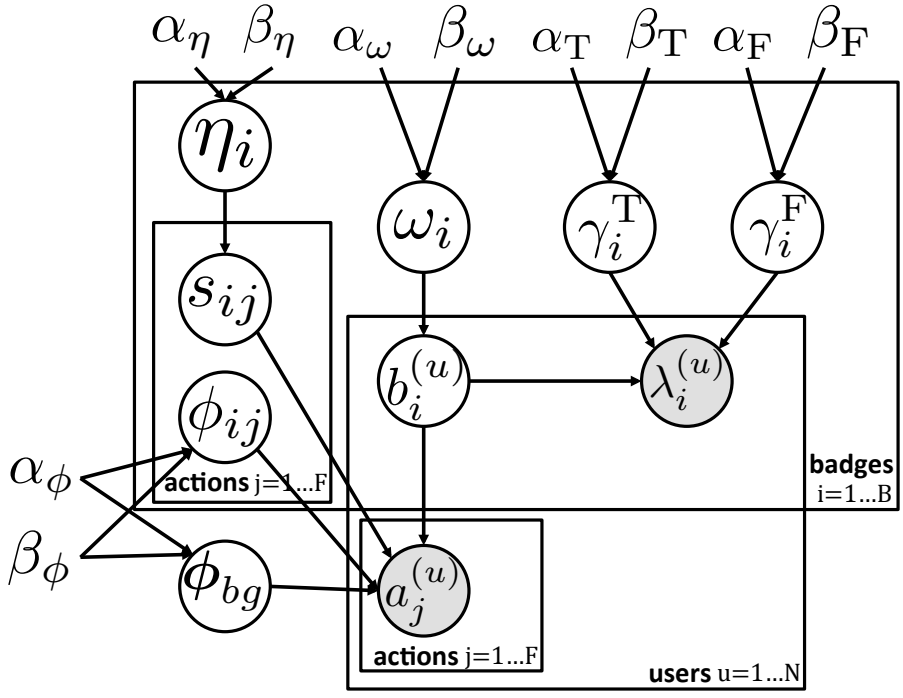
Keeping with a proper Bayesian approach, we specify prior distributions on our badge assignments $\boldsymbol{b}$, rates $\boldsymbol{\gamma}^{\mathrm{T}}$ and $\boldsymbol{\gamma}^{\mathrm{F}}$, sparsity masks $\mathbf{s}$ and action probabilities $\boldsymbol{\phi}$ encoding our modeling assumptions.

First, as some badges are more prevalent than others (e.g., there are likely more vegetarians on Twitter than machine learning enthusiasts), we assume that each badge assignment $b_i^{(u)}$ is drawn from a beta-distributed prior rate ω_i, shared across all users for each badge i.

Second, to encode that we expect the false positive rate to be considerably lower than the true positive rate, we place separate beta priors on $\boldsymbol{\gamma}^{\mathrm{T}}$ and $\boldsymbol{\gamma}^{\mathrm{F}}$, setting the hyperparameters accordingly.

Third, as we want each badge i to explain only a sparse set of actions, we place badge-specific beta-distributed prior rates η_i from which we sample s_{ij}, allowing different badges to have different degrees of sparsity.

[6] We note that, by assuming the influence of the badges to be independent of each other for a particular user, we can write this "at least one" clause as the complement of a product of complements. While this assumption may be violated in practice, we posit that the computational savings we achieve by this simplification will outweigh the induced bias.



(a)

for all badges $i = 1, \ldots, B$ **do**
  $\omega_i \sim \mathsf{Beta}(\alpha_\omega, \beta_\omega)$
  $\gamma_i^{\mathrm{T}} \sim \mathsf{Beta}(\alpha_{\mathrm{T}}, \beta_{\mathrm{T}})$
  $\gamma_i^{\mathrm{F}} \sim \mathsf{Beta}(\alpha_{\mathrm{F}}, \beta_{\mathrm{F}})$
  $\eta_i \sim \mathsf{Beta}(\alpha_\eta, \beta_\eta)$
  for all actions $j = 1, \ldots, F$ **do**
    $s_{ij}|\eta_i \sim \mathsf{Bernoulli}(\eta_i)$
    $\phi_{ij} \sim \mathsf{Beta}(\alpha_\phi, \beta_\phi)$
for all actions $j = 1, \ldots, F$ **do**
  $\phi_{\mathrm{bg},j} \sim \mathsf{Beta}(\alpha_\phi, \beta_\phi)$
for all users $u = 1, \ldots, N$ **do**
  for all badges $i = 1, \ldots, B$ **do**
    $b_i^{(u)}|\omega_i \sim \mathsf{Bernoulli}(\omega_i)$
    $\lambda_i^{(u)}|b_i^{(u)}, \gamma_i^{\mathrm{F}}, \gamma_i^{\mathrm{T}} \sim \mathsf{Bernoulli}\left(b_i^{(u)}\gamma_i^{\mathrm{T}} + (1 - b_i^{(u)})\gamma_i^{\mathrm{F}}\right)$
  for all actions $j = 1, \ldots, F$ **do**
    $a_j^{(u)}|\boldsymbol{b}^{(u)}, \boldsymbol{\phi}_{\bullet j}, \mathbf{s}_{\bullet j} \sim \mathsf{Bernoulli}\left(1 - (1 - \phi_{\mathrm{bg},j}) \prod_{i:b_i^{(u)}=1} (1 - \phi_{ij} s_{ij})\right)$

(b)

Figure 5.2: Plate diagram and generative model.

Finally, we place vague beta priors on the action probabilities $\boldsymbol{\phi}$ seeking to learn these primarily from data.

A depiction of our graphical model and a summary of the full generative process can be found in Figure 5.2. (Section 2.1 in the background chapter of this thesis contains a brief introduction to probabilistic graphical models, for the unfamiliar reader.)

5.1.4 Badge inference

Given our model and the observations from each user, we wish to infer the latent badge assignments $\boldsymbol{b}$, as well as which actions are explained by each badge (and to what degree). As computing the exact posterior probabilities in a graphical model such as this is intractable, we employ Markov Chain Monte Carlo (MCMC) methodology and estimate the posterior probabilities on $\boldsymbol{b}$, $\mathbf{s}$ and $\boldsymbol{\phi}$ by deriving a Gibbs sampler with interleaved Metropolis-Hastings steps. In particular, we derive a *collapsed* Gibbs sampler, marginalizing out $\boldsymbol{\eta}$, $\boldsymbol{\omega}$, $\boldsymbol{\gamma}^{\mathrm{T}}$ and $\boldsymbol{\gamma}^{\mathrm{F}}$, leaving only the variables of interest. This results in the following sampler:

1. *Sample* $\boldsymbol{b}$. We sample each badge assignment $b_i^{(u)}$ for a particular user u from its conditional distribution, which we can write as proportional to the product of an action likelihood, a label likelihood and a prior:

$$p(b_i^{(u)}|\boldsymbol{a}^{(u)}, \boldsymbol{\lambda}_i, \boldsymbol{b}_i^{(-u)}, \boldsymbol{\phi}, \mathbf{s}, \boldsymbol{b}_{-i}^{(u)}) \propto \\ p(\boldsymbol{a}^{(u)}|\boldsymbol{b}^{(u)}, \boldsymbol{\phi}, \mathbf{s}) \cdot p(\boldsymbol{\lambda}_i|\boldsymbol{b}_i) \cdot p(b_i^{(u)}|\boldsymbol{b}_i^{(-u)}). \tag{5.1}$$

2. *Sample* $\mathbf{s}$. We sample the binary variable s_{ij} from its conditional distribution, which we can write as follows:

$$p(s_{ij}|\boldsymbol{a}_j, \mathbf{s}_{(-ij)}, \boldsymbol{\phi}, \boldsymbol{b}) \propto p(s_{ij}|\mathbf{s}_{i(-j)}) \cdot p(\boldsymbol{a}_j|\boldsymbol{\phi}, \mathbf{s}, \boldsymbol{b}), \tag{5.2}$$

   which is a product of a prior on s_{ij} and an action likelihood term. In practice, for statistical efficiency (and following Fox [2009]), rather than sampling from the conditional distribution directly, we employ a Metropolis Hastings step with a deterministic proposal of flipping s_{ij} from some value s to its complement, $\bar{s}$ [Frigessi et al., 1993, Liu, 1996].

3. *Sample* $\boldsymbol{\phi}_i$. To sample ϕ_{ij}, we first write its conditional distribution as a product of a prior and an action likelihood term:

$$p(\phi_{ij}|\boldsymbol{a}_j, \boldsymbol{\phi}_{-(ij)}, \mathbf{s}, \boldsymbol{b}) \propto p(\phi_{ij}) \cdot p(\boldsymbol{a}_j|\boldsymbol{\phi}, \mathbf{s}, \boldsymbol{b}). \tag{5.3}$$

   We use a Metropolis Hastings step here to obtain our sample, with a beta-distributed proposal distribution:

$$q(\phi'_{ij}|\phi_{ij} = \phi) = \mathsf{Beta}(\phi'_{ij}; \phi\nu, (1-\phi)\nu), \tag{5.4}$$

   parameterized with mean ϕ and effective sample size ν, meaning that each proposal is centered around the ϕ of the previous step.

4. *Sample* $\boldsymbol{\phi}_{\mathrm{bg}}$. We sample the background action probability, $\phi_{\mathrm{bg},j}$, in the same manner that we sample the per badge action probability, using a Metropolis Hastings step with the same proposal distribution.

Further details of our sampling algorithm, including complete derivations of all conditional distributions, can be found in Appendix 5.5, at the end of this chapter.

Table 5.1: The 31 badges we defined for our experiments, as specified by their corresponding labels.

1	vegetarian	17	entrepreneur
2	Apple fanboy	18	golfer
3	cyclist	19	wine lover
4	gamer	20	book worm
5	runner	21	coffee
6	hacker	22	Harry Potter
7	feminist	23	Ruby on Rails
8	photographer	24	Manchester United
9	teacher	25	Hello Kitty
10	artist	26	anime
11	foodie	27	Warcraft
12	hipster	28	jetsetter
13	NASCAR	29	Taylor Swift
14	redneck	30	Lady Gaga
15	country music fan	31	jQuery
16	yoga		

5.2 Experimental Results

5.2.1 Data

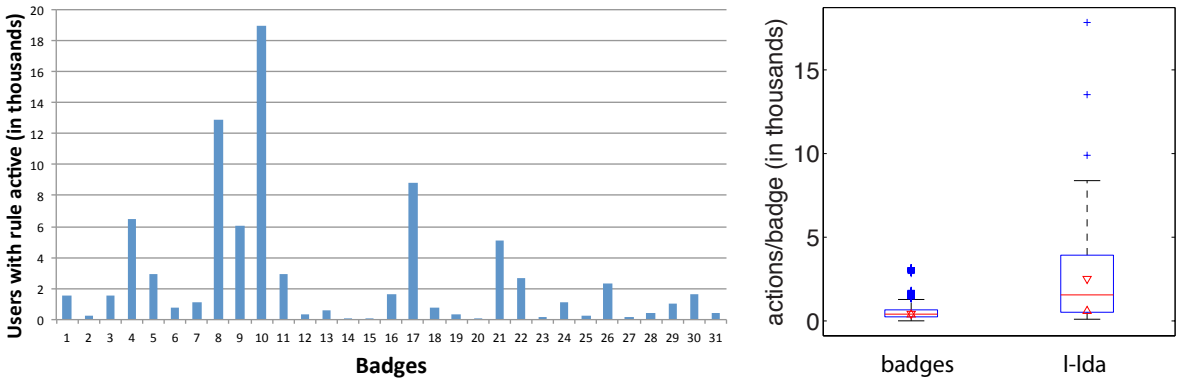
We evaluate our model on a data set of approximately seven million active Twitter users by monitoring Twitter's "firehose" stream in early August 2011, recording users with non-empty profiles. We scanned through these seven million users–which at the time of collection represented approximately 3.5 percent of all Twitter users–and manually defined a set of 31 badges by specifying a label for each one, based on the occurrence of a particular phrase or word in each user's Twitter profile. For example, we define a "vegetarian" badge by specifying:

$\lambda^{(u)}_{\text{vegetarian}} = 1$ *if user* u *has the word "vegetarian" in her Twitter profile.*

Table 5.1 contains a full listing of our 31 badges. We note that while these badges were defined as a proof of concept, a real personalization system would include a principled method for defining a large quantity of badges, as discussed further in Section 5.4 and Chapter 6.

Of the seven million users in our data set, we identified 376,916 that had at least one of the 31 labels in their profiles. We took this subset of users and monitored the firehose for one week, from 5 August 2011 to 12 August 2011, collecting every tweet and retweet by these users. This resulted in a set of approximately two million tweets. From these tweets, we extracted all unique hashtags (e.g., `#runkeeper`) and retweeted users (e.g., `@MacRumors`), defining a vocabulary of actions. We culled this vocabulary to remove any actions belonging to just a single user, leaving us with a final vocabulary of 32,030 actions (broken down into 18,003 hashtags and 14,027 retweets), performed by 75,880 different users. The most common action over this time period was the hashtag `#londonriots`, referring to the riots that took place in the British capital during the week of our data collection. Moreover, Figure 5.3a shows how many users in the data set described themselves using each of the 31 labels, with the most common being "artist."

Finally, we note that our model is not dependent on Twitter, as both the labels and the actions could be



(a) Number of users with each label present in their Twitter profile

(b) Sparsity

Figure 5.3: (a) A bar chart indicating how many users in our Twitter data set have each of the 31 labels (corresponding to each of our badges.) (b) A box plot showing that the badge definitions we learn are significantly sparser than the topics learned by labeled LDA. The horizontal line in the middle of each box represents the median number of actions per badge, and the triangles delineate confidence intervals, giving us a 5% significance threshold.

defined to take advantage of any other user signals one has access to, including location, shopping patterns, clicks, query logs and so on.[7] Twitter is, however, a convenient open platform for experimentation.

5.2.2 Evaluation

We ran our sampler on the data set described above, estimating posterior probabilities of badge assignments ($\boldsymbol{b}$) and badge definitions (ϕ and $\mathbf{s}$) under our modeling assumptions. For each iteration, our sampler has time complexity $O\left(B\left(F+N\right)\right)$. Our implementation in the F# functional programming language achieves a runtime of approximately 3.5 minutes per sample, which is the time it takes to make a single, complete pass over more than 3.3 million random variables. Our hyperparameter settings and initialization condition are detailed in Appendix 5.6.

In an effort to compare our methodology to a state-of-the-art alternative, we wanted a model that: (1) can represent multiple labels per user; (2) provides a mechanism for interpreting the definitions of each badge; and, (3) can probabilistically infer badge assignments, especially in cases where the corresponding label is not present for the particular user.

We found the most suitable comparison technique to be the labeled latent Dirichlet allocation (labeled LDA) model of Ramage et al. [2009]. This model makes a slight, but important, modification to the original LDA model, by assuming that each document is labeled with one or more tags, with each tag associated with one and only one topic. Thus, e.g., a document labeled with tags 1, 2 and 5 is assumed to have been generated from topics 1, 2 and 5, and no others. Like our model, labeled LDA provides a level of identifiability not obtained in traditional unsupervised approaches.

[7]For instance, we can imagine generalizing "labels" to instead represent any high-precision, low-recall *rule* that we intend on associating with a badge, not necessarily based on the content of a user's Twitter profile.

In order to adapt labeled LDA to our setting, we first associate a tag (and thereby a topic) with each badge, as well as an additional tag corresponding to a background topic. In our particular example, this leads to 32 unique tags. We then run labeled LDA twice: once for learning badge definitions, and once for inferring badge probabilities for each user.[8] This two-stage approach contrasts with our model–which performs both functions simultaneously–and is necessary here because the labeled LDA model does not allow us to specify uncertainty in the label assignments.

In the first run of labeled LDA, we assign each user tags corresponding to the labels present in his or her user profile, as well as the background tag. For example, a user with the word "runner" in his Twitter profile would have to be modeled by only two topics: the one corresponding to the "runner" label, and the background topic. This first run learns a topic corresponding to each of the badges, giving the probability that each badge explains each action in our vocabulary. However, as each topic is a multinomial distribution over actions, its probabilities must sum to one, leading to qualitative differences with the badges learned from our model. First, for a given badge i in our model, the probability of each action ϕ_{ij} lies in the set $(0, 1)$, and are conditionally independent of each other given their prior. This allows several actions to have high probability of being explained by the same badge, if that is what can best model the user data. Second, by explicitly modeling sparsity using the $\mathbf{s}$ variables, a badge is not forced to explain actions it is only weakly associated with, simply for the sake of getting its distribution to sum to one. This is a characteristic not only of labeled LDA, but of all such topic models. Figure 5.3b shows the difference in the sparsity of our badge definitions when compared to labeled LDA.

After learning the topics with the first run of labeled LDA, we keep them fixed and infer the badge assignments, this time giving all 32 tags to each user, allowing for badges to be inferred beyond the ones corresponding to observed labels. However, as before, because labeled LDA provides no explicit model of sparsity, and is modeling a multinomial distribution, every user will, in expectation, be assigned to one badge, but this probability mass will be spread over all 32 topics, even if they are all unlikely.

We take the badges learned and inferred by our model and compare it to those from labeled LDA, evaluating both the interpretability of the badge definitions and the correctness of the badge assignments.

Interpretability of Badge Definitions

One important desideratum from our problem description is that, whichever model we use, if we are to bring transparency to the personalization process, we must provide users with meaningful and interpretable answers when they ask, "Why did I get badge X?" A convenient way to visualize badge definitions is via word clouds, with the size of an action proportional to its weight in the badge, concisely summarizing what it means to have a particular badge. Specifically, in our model, the "weight" of action j in badge i refers to the quantity $s_{ij}\phi_{ij}$, while in labeled LDA, the weight is the probability of action j coming from topic i.

Figure 5.4 shows six examples of badges learned from running our model on the Twitter data set described above.[9] These badges do an excellent job of describing actions that are consistent with their definitions, and are precisely the types of explanations we would hope to expose to the user. For instance, the "runner" badge in Figure 5.4d explains the action `#runkeeper`, which is a hashtag automatically tweeted when

[8] In both cases, we use the implementation provided by Ramage and Rosen in the Stanford Topic Modeling Toolbox, using the default hyperparameter settings and the CVB0 inference algorithm. (`http://nlp.stanford.edu/software/tmt/`)

[9] A full visualization of all 31 badges learned from our model can be found in Appendix 5.7.

(a) Harry Potter (b) Vegetarian

(c) Manchester United (d) Runner

(e) Entrepreneur (f) Redneck

Figure 5.4: Word clouds (generated via `www.wordle.net`) representing six (of 31) badges learned by our model. Here, the size of a word is proportional to the action probability ϕ_{ij}.

using a particular smartphone application[10] that helps manage and track a user's workouts.

However, looking at Figure 5.4f, which we learn by generalizing from the actions of self-described "rednecks," we find that while some actions are expected for such a badge (e.g., #`teaparty` and #`tcot`[11]), others are more surprising, (e.g., #p2, a popular hashtag among progressives). In fact, looking at other hashtags here such as #`obama`, #`debt` and #`syria`, we see that we actually learn a more general badge, referring to American politics and global affairs, rather than one narrowly focused on rednecks. A plausible explanation for this phenomenon can be found in Figure 5.3a, where we see that "redneck," associated with label 14, appears in the Twitter profiles of very few users–perhaps too few to effectively learn what it means to be a "redneck" on Twitter.

Figure 5.5 shows two other badges corresponding to labels present in very few users. The first example, Figure 5.5a, is a more extreme form of overgeneralization than the "redneck" badge, as we see the idea of a wine lover translating to actions representing enjoyable (and often expensive) interests and activities, such as #`swarowski`, #`travel` and #`jewelry`. Figure 5.5b, on the other hand, shows the extreme situation where the original label–in this case for "Ruby on Rails"–is present in so few users that the badge can be completely overwhelmed by a more popular topic. Here, we see that this badge has been taken by actions relating to the London riots, which was the most prevalent news item in our data.

[10] `http://www.runkeeper.com`

[11] "Top Conservatives on Twitter"

When we look at the topics learned by labeled LDA, we find that they also represent interpretable badge descriptions. However, as we described earlier in this section, we do not find the same sparsity that we achieve using our model, because topic modeling approaches assume each topic is a distribution over the entire vocabulary. This contrast is made clear in Figure 5.6, where we see an extremely sparse badge representation for Apple fanboys, as learned by our model, compared to a much denser distribution over actions, learned by labeled LDA. The badge we learn focuses on a few informative actions, such as following `@MacRumors`, whereas the topic learned using labeled LDA includes, in addition to many Apple-related actions, many actions that are just tangentially related (e.g., `#runkeeper`).

Correctness of Badge Assignments

The fundamental goal of this work is not just to produce interpretable badges, but to accurately assign badges to users based on their actions. After all, badges are only useful for personalization if we infer them correctly. We expect our model to significantly outperform labeled LDA here, as we explicitly model the uncertainty relating badges and their corresponding labels, which labeled LDA does not.

In order to quantitatively measure our performance in this area, and compare it to that of labeled LDA, we re-trained both models on the Twitter data set, this time holding out a random tenth of present labels, which we treat as ground truth labels that we seek to recover. Specifically, of all badge-user pairs (i, u) corresponding to present labels $\lambda_i^{(u)} = 1$ (of which there are 83,020), we select 10% uniformly at random, and hold them out. We then take the perturbed data set and run both labeled LDA (two stages, as before) and our model, leading to estimated posterior probabilities of badge assignments, $b_i^{(u)}$.

In order to compare the two models as fairly as possible, we take each user and rank his or her badges from most probable to least probable, and see where the held out points (i, u) appear. When ranking badges for a user u using our model, we rank them by descending posterior probability of $b_i^{(u)} = 1$. When ranking based on the labeled LDA model, we rank the topics for each user by decreasing topic proportion. If label i was held out for user u, then the better of the two models will rank badge i closer to the top.[12] Figure 5.7a demonstrates that our model significantly outperforms labeled LDA on this metric, with the held out badges appearing, on average, approximately four positions higher in the ranking. Moreover, we hypothesize that the more active a user is (i.e., the more actions we observe for her in our data set), the better we will do in predicting the held out badge, as we will have more information to base our inference on. This prediction is confirmed in Figure 5.7b, where we see a six position difference in the ranking separating the most active from the least active users.

Moving beyond this quantitative comparison, we observe several qualitative properties of our inferred badges that provide anecdotal support for our model's effectiveness. First, as we model each user's badge assignments as a binary vector, we can use our samples to estimate the posterior marginal probability on *pairs* of badges appearing together in the same user, as predicted by our model. These results, shown in the annotated matrix in Figure 5.8, indicate that the hot spots of badge co-occurrence correspond to pairs of badges that we would expect to see together. In particular, the top four pairs of badges, ranked by decreasing posterior probability, are:

1. Entrepreneur and jQuery
2. Feminist and "London riots" (originally the "Ruby on Rails" badge)

[12] We note that this is a fairer comparison than directly measuring the posterior badge assignment probability, since the labeled LDA probabilities are constrained to sum to one, giving us an unfair advantage. By ranking the badges, we avoid this problem.

(a) Wine lover: overgeneralization

(b) Ruby on Rails: overwhelmed

Figure 5.5: Two cases showing current limitations of our model that arise when we have labels present in only a small number of people.

(a) badges

(b) labeled LDA

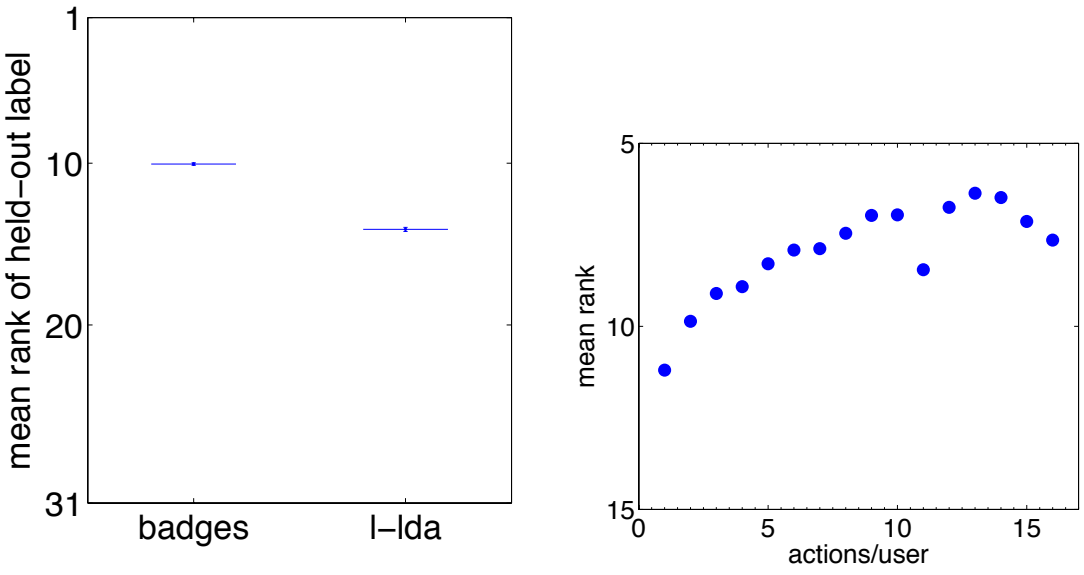
Figure 5.6: Two word clouds for the "Apple fanboy" badge, contrasting the sparsity of our learned badge (a) versus the corresponding labeled LDA topic (b).

3. Feminist and "American politics" (the generalized "redneck" badge)
4. Photographer and "London riots."

The matrix shows many other correspondences, for instance, tying together pop icons Taylor Swift and Lady Gaga.

Another view of our inferred badge assignments can be obtained by selecting a population of Twitter users from our data set, and visualizing their collective badge profile in aggregate. Here, the size of a badge label in each word cloud is proportional to its mean posterior probability across the entire population of interest. For example, we can compute the mean probability that someone with the word "conservative" in his Twitter profile is assigned the "entrepreneur" badge. Figure 5.9a shows such a word cloud for precisely this population of conservative Twitter users from our data set. We see a heavy focus on the redneck badge among this population, since it's plausible that those with the word "conservative" in their Twitter profiles are apt to tweet about American politics. Interestingly, this is true even though conservatives are more likely to describe themselves as "teachers" or "entrepreneurs", as shown in Figure 5.9b, potentially indicating that an interest in politics plays a larger role in influencing Twitter behavior than, say, being a teacher.

Finally, we can compare the inferred badge profiles of two different populations, and look at the differences between them. Naturally, to contrast with the "conservative" population of Twitter users, we also gather all users in our data set with the word "liberal" in their Twitter profile. We can than take our estimate of



(a) (b)

Figure 5.7: Results showing that, on a held-out set of labels, **(a)** our algorithm is better able to recover (approximately) ground truth badges than labeled LDA, and, **(b)** more active users get better predictions. (Error bars indicating standard error are too small to be visible.)

the mean posterior probability for each badge in each of the two populations, and visualize the difference between them, which we display in Figures 5.9c (conservatives - liberals) and 5.9d (liberals - conservatives). For example, if we look at the word "feminist" in Figure 5.9d, its size is proportional to our estimate of the mean posterior difference, $p(b_{\texttt{feminist}}^{(\text{liberals})}) - p(b_{\texttt{feminist}}^{(\text{conservatives})})$. As expected, we see, e.g., that the "feminist" badge is the one that is most likely to occur in a liberal and not a conservative.

5.3 Related Work

Since Zaslow's article on TiVo was published in 2002 [Zaslow, 2002], many forms of personalization have appeared or intensified on the Web, and they operate with varying degrees of transparency:

- The most transparent and interpretable personalization today is arguably done by Amazon.com. A user gets a selection of "recommended for you" items, with each being associated with a "because you purchased" explanation. If the recommendation is questionable, a user is allowed to correct the system by selecting a "because you purchased" item and indicating "don't use for recommendations." Amazon's feedback is similar to a user revealing the true value of an inferred badge in our model, but more specific. Furthermore, their item-to-item personalization holds the advantage that only user feedback is revealed in any explanations. The reliance on a well-represented catalogue is not feasible in many scenarios, though, and this is most notable where content is user-generated.
- Pandora[13] provides music recommendation based on likes and dislikes of each user, grounded in a

[13] `http://www.pandora.com`

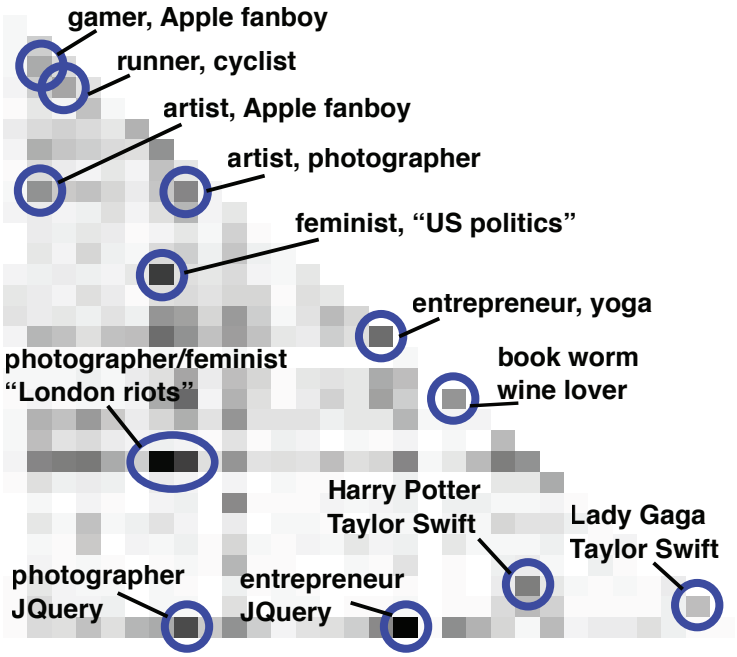


Figure 5.8: A matrix depicting the posterior marginal probability of badge co-occurrence, as estimated by our model. Darker squares represent higher probabilities. The badges are ordered in the same manner as presented in Table 5.1.

“deeply detailed, hand-built musical taxonomy” known as the Music Genome Project.[14] Pandora’s personalization service is quite transparent, with each song recommendation accompanied by an explanation for why it was selected, making clear the connection between user feedback and personalization. However, again, a hand-curated feature set like Pandora’s is problematic to maintain for more dynamically generated content.

- Google provides users with a Privacy Center,[15] from which they can opt out of having their Web and search history tracked, which affects personalization of the search results. Additionally, Google has recently allowed its users to see the attributes that it has inferred about them for ad prediction (e.g., “Demographics - Age - 65+”), giving them the option to decline any incorrect or undesired attributes.[16] While in the spirit of what we propose in this chapter, this particular dashboard is focused on addressing the quality of personalized advertising, which is not the primary reason users interact with Google. At the time of writing, there does not seem to be any such window for viewing how search results are personalized.

- Bing search[17] is personalized based on search history when signed in with a Microsoft Live account; users see “Search history has changed the ranking of these results. Learn more.” on the bottom of their search results. Clicking on “Learn more” provides an opportunity to toggle on or off the personalization, as well as information about how to remove some or all of the search from search

[14] http://blog.pandora.com/press/pandora-company-overview.html
[15] http://www.google.com/intl/en/privacy/
[16] http://googleblog.blogspot.com/2011/10/increasing-transparency-and-choice-with.html
[17] http://www.bing.com

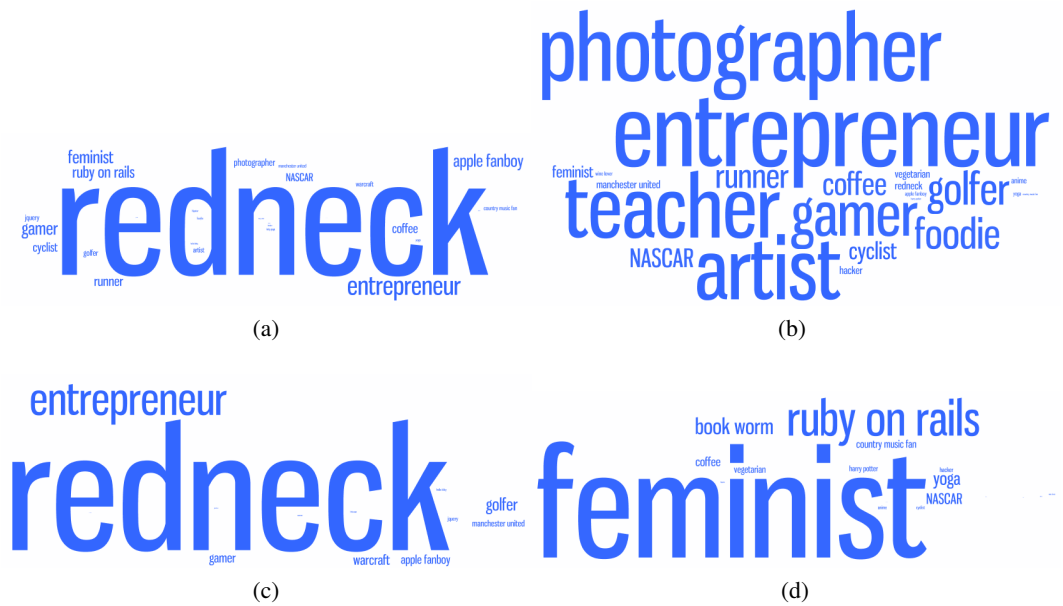


Figure 5.9: These four word clouds depict the badges learned for two populations of people: those with the word "conservative" in their Twitter profiles (254 such users in our data set), and those with the word "liberal" in their profiles (327 of them). (As these clouds represent badges (and not words or actions from our vocabulary, as the other clouds in this chapter do), we color them in blue, to differentiate them for the reader.) **(a)** Shows the mean badge probabilities for self-described conservatives. The size of a badge name is proportional to the mean probability of that badge as estimated by our model. **(b)** Shows the proportion of "conservatives" that have each of the 31 labels present in their profiles. **(c)** These badges are more likely for self-described conservatives than self-described liberals, and the size of each badge label is proportional to the absolute difference ($p(b_i^{(\text{conservatives})}) - p(b_i^{(\text{liberals})})$). **(d)** Badges more likely for liberals than conservatives. (Recall from Figure 5.5b that the "Ruby on Rails" badge is instantiated as a "London riots" badge.)

history, but it is non-transparent how this affects personalization.

- Social networks such as Facebook and Google+ are (quite openly) repositories of structured personal information, and thus the primary transparency issue with these sites is not what information they infer about their users, but rather how it is used for personalization. For example, at the time of writing, Facebook's News feed is divided into a personalized list of "Top stories" and a timely list of "Recent stories," but there is no way for a user to see why a particular status update appeared in one feed and not the other.

While this is certainly not an exhaustive list, and personalization capabilities are routinely being added and removed from these sites and others, the websites mentioned represent a large portion of user interaction on the Web (e.g., Google and Bing search amounted to nearly 85 percent of all web searches from the United States in November 2012,[18] while nearly 600 million users log on to Facebook per day at the time of writing,[19]) demonstrating the importance of studying methods for making personalization more

[18] http://www.comscore.com/Insights/Press_Releases/2012/12/comScore_Releases_November_2012_U.S._Search_Engine_Rankings

[19] http://newsroom.fb.com

transparent.

From the perspective of methodology, personalization by inferring latent user features has come a long way. While collaborative filtering through factorizing a user-item matrix, and variants thereof, is extremely successful as a backbone of recommender systems [Koren et al., 2009], the latent features don't have any interpretable meaning. Interpretability often means discretization. In this vein, Porteous et al. extended matrix factorization with discrete class allocations [2010], and although we are not aware of its existence, it is entirely feasible to enforce labels to certain class allocations.

When the domain of interest includes user-generated content, like blogs and tweets, latent topic models have frequently provided a successful modeling framework. Unfortunately, as is the case with matrix factorization techniques, the topics learned in such unsupervised models are not identifiable, a problem we avoid by associating each badge with a label. Topic models that incorporate supervision at the topic level, such as labeled LDA, explained in Section 5.2.2, and the more recent work by Andrzejewski et al. [2011], provide a mechanism for such identifiability. For instance, labeled LDA was recently used to model another axis of personalization on Twitter, mapping posts as informative, personal status updates, or inter-user social communication [Ramage et al., 2010].

5.4 Conclusions

In this chapter we have presented a Bayesian inference algorithm to learn both a descriptive model and predictor for *badges* based on user activity on a microblogging site (Twitter). In our work, a badge is seeded by a label—more generally, a high precision, low recall rule—based on self-reported user information, while our predictive model for badges explicitly relies on the presence or absence of user actions. Both these modeling decisions contribute greatly to the transparency of the prediction and we believe transparency will be a critical building block for personalization systems that users will find acceptable and suitable.

We have shown empirically that our model outperforms state-of-the-art models such as labeled LDA in terms of predictive performance, while producing interpretable descriptions of badges.

However, there are a number of open questions and challenges yet to be addressed:

- **Scale**: The current inference algorithm is a combination of exact inference (*collapsing* some of the conditional priors; see Section 5.1.4) and Gibbs sampling with interleaved Metropolis-Hastings steps. The latter is applied for all B badges and F actions. This can become problematic if the number of actions grows unbounded, e.g., if an action is the presence of a word in a status update. In order to scale this approach to real-time streams, a single-pass approximate inference algorithm needs to be developed, with special attention paid to inference algorithms conducted in a distributed setting (cf. [Ahmed et al., 2012a, Gonzalez et al., 2012, Low et al., 2012, Smola and Narayanamurthy, 2010]).
- **Dynamics**: Currently, our model assumes a fixed but unknown dependency on user actions and badges. In a more realistic setting, this dependency will vary over time.
- **Badge definitions**: As badges are defined simply by indicating a label, and in a realistic setting, a large number of them will be needed to model the full spectrum of user behavior, it is imperative to consider approaches for automatically defining a rich set of badges, thus scaling our model to be more expressive. In order to maintain a desired level of human interpretability, it may be necessary to keep humans in the loop when defining new badges, and thus one approach might be to crowd-source

badge definitions. In this case, the primary open question revolves around how to appropriately incentivize the creation of high quality badges.

In the next chapter, we will build upon this work to create a badge-based concept representation that addresses these three open problems. Specifically, now that we have a transparent model of user behavior, we will investigate a relationship between badges and concepts, allowing us to come full circle and use badges in our interactive concept coverage framework.

5.5 Appendix: Derivations

We only sample $b_i^{(u)}$, $\phi_{\text{bg},j}$, ϕ_{ij} and s_{ij}, while collapsing everything else. We assume that $a_j^{(u)}$ and $\lambda_i^{(u)}$ are observed.

5.5.1 Sampling $b_i^{(u)}$

$$
\begin{aligned}
P(b_i^{(u)}|\mathbf{a}^{(u)}, \boldsymbol{\lambda}_i, \mathbf{b}_i^{(-u)}, \boldsymbol{\phi}, \mathbf{s}, \mathbf{b}_{-i}^{(u)}) & && (5.5)\\
\propto P(\mathbf{a}^{(u)}, \boldsymbol{\lambda}_i|\mathbf{b}_i, \mathbf{b}_{-i}^{(u)}, \boldsymbol{\phi}, \mathbf{s}) P(b_i^{(u)}|\mathbf{b}_i^{(-u)}) & && (5.6)\\
= P(\mathbf{a}^{(u)}|\mathbf{b}_i, \mathbf{b}_{-i}^{(u)}, \boldsymbol{\phi}, \mathbf{s}) P(\boldsymbol{\lambda}_i|\mathbf{a}^{(u)}, \mathbf{b}_i, \mathbf{b}_{-i}^{(u)}, \boldsymbol{\phi}, \mathbf{s}) P(b_i^{(u)}|\mathbf{b}_i^{(-u)}) & && (5.7)\\
= \underbrace{P(\mathbf{a}^{(u)}|\mathbf{b}^{(u)}, \boldsymbol{\phi}, \mathbf{s})}_{A} \underbrace{P(\boldsymbol{\lambda}_i|\mathbf{b}_i)}_{B} \underbrace{P(b_i^{(u)}|\mathbf{b}_i^{(-u)})}_{C} & && (5.8)
\end{aligned}
$$

On line 5.8, A and B correspond to the likelihood contributions of $\mathbf{a}^{(u)}$ and $\boldsymbol{\lambda}_i$, respectively. C corresponds to the prior term on $b_i^{(u)}$. We take these each in turn.

A:

$$
\begin{aligned}
P(\mathbf{a}^{(u)}|\mathbf{b}^{(u)}, \boldsymbol{\phi}, \mathbf{s}) &= \prod_{j=1}^{F} P(a_j^{(u)}|\mathbf{b}^{(u)}, \boldsymbol{\phi}_{\bullet j}, \mathbf{s}_{\bullet j}) && (5.9)\\
&\propto \prod_{j:s_{ij}=1} P(a_j^{(u)}|\mathbf{b}^{(u)}, \boldsymbol{\phi}_{\bullet j}, \mathbf{s}_{\bullet j}) && (5.10)\\
&= \prod_{j:s_{ij}=1} \mathsf{Bernoulli}(a_j^{(u)}; 1 - (1 - \phi_{\text{bg},j}) \prod_{k:b_k^{((u))}=1} (1 - \phi_{kj} s_{kj})) && (5.11)
\end{aligned}
$$

B:

$$P(\boldsymbol{\lambda}_i|\mathbf{b}_i) = \int_{\gamma_i^T}\int_{\gamma_i^F} P(\boldsymbol{\lambda}_i, \gamma_i^T, \gamma_i^F|\mathbf{b}_i)d\gamma_i^T d\gamma_i^F \tag{5.12}$$

$$= \int_{\gamma_i^T}\int_{\gamma_i^F} P(\boldsymbol{\lambda}_i|\gamma_i^T, \gamma_i^F, \mathbf{b}_i)P(\gamma_i^T, \gamma_i^F|\mathbf{b}_i)d\gamma_i^T d\gamma_i^F \tag{5.13}$$

$$= \int_{\gamma_i^T}\int_{\gamma_i^F} \left(\prod_{u=1}^{N} P(\lambda_i^{(u)}|\gamma_i^T, \gamma_i^F, b_i^{(u)})\right) P(\gamma_i^T)P(\gamma_i^F)d\gamma_i^T d\gamma_i^F \tag{5.14}$$

$$= \int_{\gamma_i^T}\int_{\gamma_i^F} (\gamma_i^T)^{m_{i+}}(1-\gamma_i^T)^{m_{i-}}(\gamma_i^F)^{\overline{m}_{i+}}(1-\gamma_i^F)^{\overline{m}_{i-}} P(\gamma_i^T)P(\gamma_i^F)d\gamma_i^T d\gamma_i^F \tag{5.15}$$

$$\propto \int_{\gamma_i^T}\int_{\gamma_i^F} (\gamma_i^T)^{m_{i+}+\alpha_T-1}(1-\gamma_i^T)^{m_{i-}+\beta_T-1}(\gamma_i^F)^{\overline{m}_{i+}+\alpha_F-1}(1-\gamma_i^F)^{\overline{m}_{i-}+\beta_F-1}d\gamma_i^T d\gamma_i^F \tag{5.16}$$

$$= B(m_{i+}+\alpha_T, m_{i-}+\beta_T)B(\overline{m}_{i+}+\alpha_F, \overline{m}_{i-}+\beta_F) \tag{5.17}$$

$$= \frac{\Gamma(m_{i+}+\alpha_T)\Gamma(m_{i-}+\beta_T)}{\Gamma(\alpha_T+\beta_T+m_{i+}+m_{i-})}\frac{\Gamma(\overline{m}_{i+}+\alpha_F)\Gamma(\overline{m}_{i-}+\beta_F)}{\Gamma(\alpha_F+\beta_F+\overline{m}_{i+}+\overline{m}_{i-})} \tag{5.18}$$

$$= \frac{\Gamma(m_{i+}+\alpha_T)\Gamma(m_{i-}+\beta_T)\Gamma(\overline{m}_{i+}+\alpha_F)\Gamma(\overline{m}_{i-}+\beta_F)}{\Gamma(\alpha_T+\beta_T+m_i)\Gamma(\alpha_F+\beta_F+N-m_i)}, \tag{5.19}$$

where:

- m_{i+} is the number of users **with** badge i and **with** rule i,
- m_{i-} is the number of users **with** badge i but **not** rule i,
- $\overline{m}_{i+}$ is the number of users **without** badge i but **with** rule i,
- $\overline{m}_{i-}$ is the number of users **without** badge i and **without** rule i, and,
- m_i is the total number of users with badge i.

Thus, if we are sampling $b_i^{(u)}$ where $\lambda_i^{(u)} = 1$, the expression in (5.19) takes the following value for $b_i^{(u)} = 0$:

$$\frac{\Gamma(m_{i+}^{(-u)}+\alpha_T)\Gamma(\overline{m}_{i+}^{(-u)}+\alpha_F+1)}{\Gamma(\alpha_T+\beta_T+m_i^{(-u)})\Gamma(\alpha_F+\beta_F+N-m_i^{(-u)})}, \tag{5.20}$$

and this value for $b_i^{(u)} = 1$:

$$\frac{\Gamma(m_{i+}^{(-u)}+\alpha_T+1)\Gamma(\overline{m}_{i+}^{(-u)}+\alpha_F)}{\Gamma(\alpha_T+\beta_T+m_i^{(-u)}+1)\Gamma(\alpha_F+\beta_F+N-m_i^{(-u)}-1)}, \tag{5.21}$$

where the $(-u)$ superscript indicates a value computed ignoring user u. We can simplify out the gamma functions by looking at the ratio of (5.20) to (5.21), which gives us,

$$\frac{\overline{m}_{i+}^{(-u)} + \alpha_F}{\alpha_F + \beta_F + N - m_i^{(-u)} - 1} \frac{\alpha_T + \beta_T + m_i^{(-u)}}{m_{i+}^{(-u)} + \alpha_T}. \tag{5.22}$$

Similarly, this same ratio for the case where $\lambda_i^{(u)} = 0$ ends up being:

$$\frac{\overline{m}_{i-}^{(-u)} + \beta_F}{\alpha_F + \beta_F + N - m_i^{(-u)} - 1} \frac{\alpha_T + \beta_T + m_i^{(-u)}}{m_{i-}^{(-u)} + \beta_T}. \tag{5.23}$$

C:

$$\begin{aligned}
P(b_i^{(u)}|\mathbf{b}_i^{(-u)}) &= \int_{\omega_i} P(b_i^{(u)}, \omega_i|\mathbf{b}_i^{(-u)}) d\omega_i && (5.24)\\
&= \int_{\omega_i} P(b_i^{(u)}|\omega_i, \mathbf{b}_i^{(-u)}) P(\omega_i|\mathbf{b}_i^{(-u)}) d\omega_i && (5.25)\\
&= \int_{\omega_i} P(b_i^{(u)}|\omega_i) P(\omega_i|\mathbf{b}_i^{(-u)}) d\omega_i && (5.26)\\
&\propto \int_{\omega_i} P(b_i^{(u)}|\omega_i) P(\mathbf{b}_i^{(-u)}|\omega_i) P(\omega_i) d\omega_i && (5.27)\\
&= \int_{\omega_i} P(\omega_i) \prod_{v=1}^{N} P(b_i^{((v))}|\omega_i) d\omega_i && (5.28)\\
&= \int_{\omega_i} \mathsf{Beta}(\omega_i; \alpha_\omega, \beta_\omega) \prod_{v=1}^{N} \mathsf{Bernoulli}(b_i^{((v))}; \omega_i) d\omega_i && (5.29)\\
&= \frac{1}{B(\alpha_\omega, \beta_\omega)} \int_{\omega_i} \omega_i^{\alpha_\omega + m_i - 1} (1 - \omega_i)^{\beta_\omega + N - m_i - 1} d\omega_i && (5.30)\\
&\propto B(\alpha_\omega + m_i, \beta_\omega + N - m_i), && (5.31)
\end{aligned}$$

where m_i is the number of total users with badge i (i.e., with $b_i^{(u)} = 1$).

5.5.2 Sampling $\phi_{\mathrm{bg},j}$

$$\begin{aligned}
&P(\phi_{\mathrm{bg},j}|\mathbf{a}_j, \boldsymbol{\phi}_j, \mathbf{s}_j, \mathbf{b}) && (5.32)\\
&\quad \propto P(\phi_{\mathrm{bg},j}) P(\mathbf{a}_j|\phi_{\mathrm{bg},j}, \boldsymbol{\phi}_j, \mathbf{s}_j, \mathbf{b}) && (5.33)\\
&\quad = \mathsf{Beta}(\phi_{\mathrm{bg},j}; \alpha_\phi, \beta_\phi) \prod_{u=1}^{N} \mathsf{Bernoulli}\left(a_j^{(u)}; 1 - (1 - \phi_{\mathrm{bg},j}) \prod_{i:b_i^{((u))}=1} (1 - \phi_{ij} s_{ij})\right) && (5.34)
\end{aligned}$$

Let's do a Metropolis Hastings step to sample from this conditional, with the following proposal:

$$\phi'_{\text{bg},j}|\phi_{\text{bg},j} = \phi \sim \mathsf{Beta}(\phi\nu, (1-\phi)\nu).$$

This is a Beta distribution parameterized by a mean $\mu = \phi$ and a "sample size" ν.

Our acceptance probability is thus:

$$\begin{aligned}
\rho =& \frac{P(\phi'_{\text{bg},j}|\mathbf{a}_j, \boldsymbol{\phi}_j, \mathbf{s}_j, \mathbf{b})}{P(\phi_{\text{bg},j}|\mathbf{a}_j, \boldsymbol{\phi}_j, \mathbf{s}_j, \mathbf{b})} \frac{Q(\phi_{\text{bg},j}|\phi'_{\text{bg},j})}{Q(\phi'_{\text{bg},j}|\phi_{\text{bg},j})} && (5.35)\\
=& \frac{\mathsf{Beta}(\phi'_{\text{bg},j}; \alpha_\phi, \beta_\phi)}{\mathsf{Beta}(\phi_{\text{bg},j}; \alpha_\phi, \beta_\phi)} \\
& \frac{\prod_{u=1}^N \mathsf{Bernoulli}\left(a_j^{(u)}; 1 - \left(1-\phi'_{\text{bg},j}\right)\prod_{i:b_i^{((u))}=1}(1-\phi_{ij}s_{ij})\right)}{\prod_{u=1}^N \mathsf{Bernoulli}\left(a_j^{(u)}; 1 - (1-\phi_{\text{bg},j})\prod_{i:b_i^{((u))}=1}(1-\phi_{ij}s_{ij})\right)} \\
& \frac{\mathsf{Beta}(\phi_{\text{bg},j}; \phi'_{\text{bg},j}\nu, (1-\phi'_{\text{bg},j})\nu)}{\mathsf{Beta}(\phi'_{\text{bg},j}; \phi_{\text{bg},j}\nu, (1-\phi_{\text{bg},j})\nu)} && (5.36)\\
=& \frac{(\phi'_{\text{bg},j})^{\alpha_\phi - 1}(1-\phi'_{\text{bg},j})^{\beta_\phi - 1}}{(\phi_{\text{bg},j})^{\alpha_\phi - 1}(1-\phi_{\text{bg},j})^{\beta_\phi - 1}} \\
& \left(\frac{1-\phi'_{\text{bg},j}}{1-\phi_{\text{bg},j}}\right)^{n_{-j}} \prod_{u:a_j^{(u)}=1} \frac{1-\left(1-\phi'_{\text{bg},j}\right)\prod_{i:b_i^{((u))}=1}(1-\phi_{ij}s_{ij})}{1-(1-\phi_{\text{bg},j})\prod_{i:b_i^{((u))}=1}(1-\phi_{ij}s_{ij})} \\
& \frac{\Gamma(\phi_{\text{bg},j}\nu)\Gamma((1-\phi_{\text{bg},j})\nu)}{\Gamma(\phi'_{\text{bg},j}\nu)\Gamma((1-\phi'_{\text{bg},j})\nu)} \frac{(\phi_{\text{bg},j})^{\phi'_{\text{bg},j}\nu - 1}(1-\phi_{\text{bg},j})^{(1-\phi'_{\text{bg},j})\nu - 1}}{(\phi'_{\text{bg},j})^{\phi_{\text{bg},j}\nu - 1}(1-\phi'_{\text{bg},j})^{(1-\phi_{\text{bg},j})\nu - 1}} && (5.37)\\
=& \frac{\Gamma(\phi_{\text{bg},j}\nu)\Gamma((1-\phi_{\text{bg},j})\nu)}{\Gamma(\phi'_{\text{bg},j}\nu)\Gamma((1-\phi'_{\text{bg},j})\nu)} \prod_{u:a_j^{(u)}=1} \frac{1-\left(1-\phi'_{\text{bg},j}\right)\prod_{i:b_i^{((u))}=1}(1-\phi_{ij}s_{ij})}{1-(1-\phi_{\text{bg},j})\prod_{i:b_i^{((u))}=1}(1-\phi_{ij}s_{ij})} \\
& \frac{(\phi_{\text{bg},j})^{\phi'_{\text{bg},j}\nu - \alpha_\phi}(1-\phi_{\text{bg},j})^{(1-\phi'_{\text{bg},j})\nu - \beta_\phi - n_{-j}}}{(\phi'_{\text{bg},j})^{\phi_{\text{bg},j}\nu - \alpha_\phi}(1-\phi'_{\text{bg},j})^{(1-\phi_{\text{bg},j})\nu - \beta_\phi - n_{-j}}}, && (5.38)
\end{aligned}$$

where n_{-j} is the number of users who do not perform action j (i.e., those with $a_j^{(u)} = 0$).

5.5.3 Sampling s_{ij}

$$P(s_{ij}|\mathbf{a}_j, s_{-(ij)}, \boldsymbol{\phi}, \mathbf{b}) \propto \underbrace{P(s_{ij}|\mathbf{s}_{i(-j)})}_{A} \underbrace{P(\mathbf{a}_j|\boldsymbol{\phi}, \mathbf{s}, \mathbf{b})}_{B}, \tag{5.39}$$

where we expand the two factors as follows:

A:

$$P(s_{ij}|\mathbf{s}_{i(-j)}) = \int_{\eta_i} P(s_{ij}, \eta_i|\mathbf{s}_{i(-j)})d\eta_i \tag{5.40}$$

$$= \int_{\eta_i} P(s_{ij}|\eta_i)P(\eta_i|\mathbf{s}_{i(-j)})d\eta_i \tag{5.41}$$

$$\propto \int_{\eta_i} P(\eta_i)\prod_{k=1}^{F} P(s_{ik}|\eta_i)d\eta_i \tag{5.42}$$

$$= \int_{\eta_i} \mathsf{Beta}(\eta_i;\alpha_\eta,\beta_\eta)\prod_{k=1}^{F} \mathsf{Bernoulli}(s_{ik};\eta_i)d\eta_i \tag{5.43}$$

$$\propto \int_{\eta_i} \eta_i^{\alpha_\eta+v_i-1}(1-\eta_i)^{\beta_\eta+F-v_i-1}d\eta_i \tag{5.44}$$

$$= B(\alpha_\eta+v_i, \beta_\eta+F-v_i), \tag{5.45}$$

where v_i is the number of actions that are active for badge i (i.e., number of $s_{ik}=1$ for some i).

B:

$$P(\mathbf{a}_j|\boldsymbol{\phi},\mathbf{s},\mathbf{b}) = \prod_{u=1}^{N} P(a_j^{(u)}|\boldsymbol{\phi},\mathbf{s},\mathbf{b}^{(u)}) \tag{5.46}$$

$$\propto \prod_{u:b_i^{(u)}=1} P(a_j^{(u)}|\boldsymbol{\phi},\mathbf{s},\mathbf{b}^{(u)}) \tag{5.47}$$

$$= \prod_{u:b_i^{(u)}=1} \mathsf{Bernoulli}\left(a_j^{(u)}; 1-(1-\phi_{\mathrm{bg},j})\prod_{k:b_k^{((u))}=1}(1-\phi_{kj}s_{kj})\right). \tag{5.48}$$

We use a Metropolis step to sample this variable, with a deterministic proposal of flipping the value of s_{ij} from s to $\bar{s}$. This gives us the following acceptance probability:

$$\rho = \frac{P(s'_{ij}|\mathbf{a}_j, s_{-(ij)}, \boldsymbol{\phi}, \mathbf{b})}{P(s_{ij}|\mathbf{a}_j, s_{-(ij)}, \boldsymbol{\phi}, \mathbf{b})} \tag{5.49}$$

$$= \frac{B(\alpha_\eta+v'_i, \beta_\eta+F-v'_i)}{B(\alpha_\eta+v_i, \beta_\eta+F-v_i)} \frac{\prod_{u:b_i^{(u)}=1}\mathsf{Bernoulli}\left(a_j^{(u)}; 1-(1-\phi_{\mathrm{bg},j})\prod_{k:b_k^{((u))}=1}\left(1-\phi_{kj}s'_{kj}\right)\right)}{\prod_{u:b_i^{(u)}=1}\mathsf{Bernoulli}\left(a_j^{(u)}; 1-(1-\phi_{\mathrm{bg},j})\prod_{k:b_k^{((u))}=1}\left(1-\phi_{kj}s_{kj}\right)\right)} \tag{5.50}$$

$$= \frac{B(\alpha_\eta+v'_i, \beta_\eta+F-v'_i)}{B(\alpha_\eta+v_i, \beta_\eta+F-v_i)} \left(\frac{1-\phi_{ij}s'_{ij}}{1-\phi_{ij}s_{ij}}\right)^{n_{i,(-j)}} \prod_{u:b_i^{(u)}=1\wedge a_j^{(u)}=1} \frac{1-(1-\phi_{\mathrm{bg},j})\prod_{k:b_k^{((u))}=1}\left(1-\phi_{kj}s'_{kj}\right)}{1-(1-\phi_{\mathrm{bg},j})\prod_{k:b_k^{((u))}=1}\left(1-\phi_{kj}s_{kj}\right)}, \tag{5.51}$$

where $n_{i,(-j)}$ is the number of users that have badge i but do not perform action j. Note that the proposal and reverse proposal terms don't appear here because it's a deterministic proposal.

We can simplify the first fraction further if we look at the two flip cases. When we flip $0 \to 1$, this fraction is:

$$\frac{\alpha_\eta + v_i^{(-j)}}{\beta_\eta + F - v_i^{(-j)} - 1},$$

where we write $v_i^{(-j)}$ to represent $v_i - s_{ij}$. Likewise, when we flip $1 \to 0$, we have:

$$\frac{\beta_\eta + F - v_i^{(-j)} - 1}{\alpha_\eta + v_i^{(-j)}}.$$

Note that in the case where we flip s_{ij} from 0 to 1, we sample the corresponding ϕ_{ij} from the prior, $\mathsf{Beta}(\alpha_\phi, \beta_\phi)$.

5.5.4 Sampling ϕ_{ij}

$$P(\phi_{ij}|\mathbf{a}_j, \boldsymbol{\phi}_{-(ij)}, \mathbf{s}, \mathbf{b}) \propto P(\phi_{ij})P(\mathbf{a}_j|\boldsymbol{\phi}, \mathbf{s}, \mathbf{b}) \tag{5.52}$$

$$\propto \mathsf{Beta}(\phi_{ij}; \alpha_\phi, \beta_\phi) \prod_{u:b_i^{(u)}=1} \mathsf{Bernoulli}\left(a_j^{(u)}; 1 - (1 - \phi_{\mathrm{bg},j}) \prod_{k:b_k^{((u))}=1} (1 - \phi_{kj}s_{kj})\right). \tag{5.53}$$

Let's do a Metropolis Hastings step to sample from this conditional, with the same proposal we used for sampling $\phi_{\mathrm{bg},j}$.

$$\phi'_{ij}|\phi_{ij} = \phi \sim \mathsf{Beta}(\phi\nu, (1-\phi)\nu).$$

Again, this is a Beta distribution parameterized by a mean $\mu = \phi$ and a "sample size" ν.

Our acceptance probability is thus:

$$\rho = \frac{P(\phi'_{ij}|\mathbf{a}_j, \boldsymbol{\phi}_{-(ij)}, \mathbf{s}, \mathbf{b})}{P(\phi_{ij}|\mathbf{a}_j, \boldsymbol{\phi}_{-(ij)}, \mathbf{s}, \mathbf{b})} \frac{Q(\phi_{ij}|\phi'_{ij})}{Q(\phi'_{ij}|\phi_{ij})} \tag{5.54}$$

$$= \frac{\mathsf{Beta}(\phi'_{ij}; \alpha_\phi, \beta_\phi)}{\mathsf{Beta}(\phi_{ij}; \alpha_\phi, \beta_\phi)} \frac{\prod_{u:b_i^{(u)}=1} \mathsf{Bernoulli}\left(a_j^{(u)}; 1-(1-\phi_{\mathrm{bg},j})\prod_{i:b_i^{((u))}=1}\left(1-\phi'_{ij}s_{ij}\right)\right)}{\prod_{u:b_i^{(u)}=1} \mathsf{Bernoulli}\left(a_j^{(u)}; 1-(1-\phi_{\mathrm{bg},j})\prod_{i:b_i^{((u))}=1}\left(1-\phi_{ij}s_{ij}\right)\right)} \frac{\mathsf{Beta}(\phi_{ij}; \phi'_{ij}\nu, (1-\phi'_{ij})\nu)}{\mathsf{Beta}(\phi'_{ij}; \phi_{ij}\nu, (1-\phi_{ij})\nu)} \tag{5.55}$$

$$= \frac{\Gamma(\phi_{ij}\nu)\Gamma((1-\phi_{ij})\nu)}{\Gamma(\phi'_{ij}\nu)\Gamma((1-\phi'_{ij})\nu)} \prod_{u:b_i^{(u)}=1 \wedge a_j^{(u)}=1} \frac{1-(1-\phi_{\mathrm{bg},j})\prod_{i:b_i^{((u))}=1}\left(1-\phi'_{ij}s_{ij}\right)}{1-(1-\phi_{\mathrm{bg},j})\prod_{i:b_i^{((u))}=1}\left(1-\phi_{ij}s_{ij}\right)} \frac{(\phi_{ij})^{\phi'_{ij}\nu-\alpha_\phi}(1-\phi_{ij})^{(1-\phi'_{ij})\nu-\beta_\phi-n_{i,(-j)}}}{(\phi'_{ij})^{\phi_{ij}\nu-\alpha_\phi}(1-\phi'_{ij})^{(1-\phi_{ij})\nu-\beta_\phi-n_{i,(-j)}}}. \tag{5.56}$$

5.6 Appendix: Experimental Details

5.6.1 Hyperparameters

In the experiments described in this chapter, we run our model with the following hyperparameter settings:

- $\alpha_\eta = 1, \beta_\eta = 999$: gives us an expected sparsity level per badge of 0.1 percent of all possible actions.
- $\alpha_\omega = 5, \beta_\omega = 25$: indicate that our intuition that a given badge will only be active in a small proportion of users.
- $\alpha_{\mathrm{T}} = 10, \beta_{\mathrm{T}} = 90, \alpha_{\mathrm{F}} = 1, \beta_{\mathrm{F}} = 1000$: encodes our assumption that rules are high precision and low recall, making it very unlikely for a user without a particular badge to activate its corresponding rule.
- $\alpha_\phi = 1, \beta_\phi = 99$: encodes the belief that, on average, we expect that when a badge explains an action (i.e., $s_{ij} > 0$), only 1 % of users with that badge will actually observe the action.

We note that we can also take the fully Bayesian approach and sample our hyperparameters, saving us having to set them in this way.

5.6.2 Initialization

We initialize the $\mathbf{s}$ and $\boldsymbol{\phi}$ variables for our sampler by assuming that active rules indicate positive examples, and looking at the actions performed by these users. For example, if we want to figure out which actions are explained by the "Apple fanboy" badge, we can look at the users with this rule active, and for each

Figure 5.10: Initialization condition for the Apple fanboy badge.

action, compute the proportion of these users that perform it. We then compare this proportion to the overall proportion for this action, across all users. If the within-badge proportion is greater, we assume this badge can explain this action, and initialize $s_{ij} = 1$ and ϕ_{ij} to the within-badge proportion. Figure 5.10 shows, for example, the initial state for the "Apple fanboy" badge as computed in this way (which can be compared to the posterior learned from our model in Figure 5.6a).

The other variables are initialized from their prior distributions.

5.7 Appendix: Badge Visualizations

In this appendix, we show visualizations of all 31 badges. In the following word clouds, the size of an action is proportional to its probability for the given badge.

(a) Vegetarian

(b) Apple fanboy

(c) Cyclist

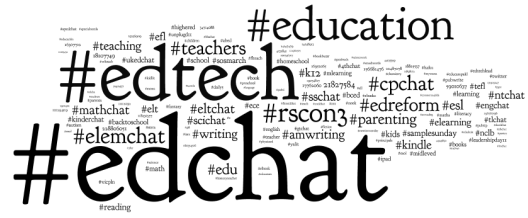
(d) Gamer

(e) Runner

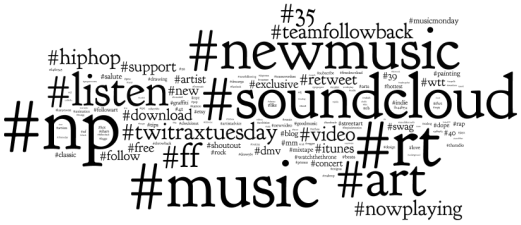
(f) Hacker

(a) Feminist

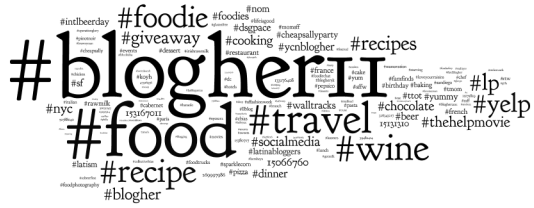
(b) Photographer



(c) Teacher



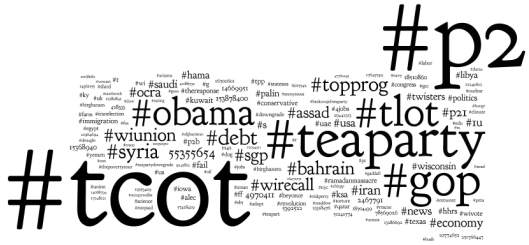
(d) Artist



(e) Foodie

(f) Hipster

(a) NASCAR



(b) Redneck

(c) Country Music Fan

(d) Yoga

(e) Entrepreneur

(f) Golfer

(a) Wine lover

(b) Bookworm

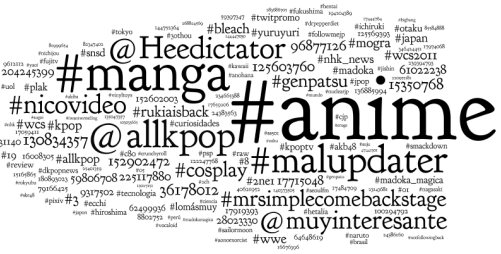
(c) Coffee

(d) Harry Potter

(e) Ruby on Rails

(f) Manchester United

(a) Hello Kitty



(b) Anime

(c) Warcraft

(d) Jetsetter

(e) Taylor Swift

(f) Lady Gaga

(g) JQuery

Chapter 6

Representing Documents Through Their Readers

In Chapter 1, we described that, in order for a computer to reason about a document's meaning, a representation must be chosen for the document's contents. So far in this thesis, we have considered simple concept representations, such as topics from a topic model [Blei and Lafferty, 2009] or individual words, named entities or noun phrases. While we were able to successfully use such concept representations in showing the effectiveness of our interactive concept coverage framework, we are still left with the question: can we do a better job representing documents?

In order to answer this question, we must consider issues that arise when using simpler document representations:

- In realistic recommendation systems, documents arrive in a stream, and thus the user feedback obtained in one time period must be used to recommend articles in another time period. As such, the preferences we learn over a document representation must be transferable across time. For simple concept representations, this can lead to a troublesome matching problem (e.g., figuring out which LDA topics in one time period match which topics in another).
- In our methodology, we model user interests as a weight vector over concepts. It would be preferable if our concepts aligned with dimensions that users might express interest in. For example, while it is conceivable that a user might be interested in a word or named entity (e.g., "Obama" or "Michigan"), it is more far-fetched for a user to have a coherent, consistent interest over a coarse-grained LDA topic.
- Simple concept representations tend to either be opaque (e.g., LDA topics) or very high dimensional (e.g., a vocabulary over words). We would like to use the ideas from the previous chapter to make personalization more transparent, but to do so, we need a concept representation that can exploit the idea of badges.
- Different users may find different granularities of concepts more appropriate. To an avid sports fan, for example, the Pittsburgh Steelers, LeBron James and Wimbledon all represent distinct concepts from the world of sports. However, another user who may not care so much about sports may consider these all to be equivalent from his point of view. Namely, for this user, an article about the Steelers might just as well be an article about tennis; it's all "just sports" to him.

In this chapter, we will address the first three issues described above, while the latter one is deferred for future work, as discussed in Chapter 8. Specifically, we will take a cue from the previous chapter, and explore how we might represent documents based on the attributes of their likely readers.

6.1 Documents and Their Readers

In today's world, it has become commonplace for readers to share news articles and blog posts with their friends and followers on social networking sites. Understanding that much of their future success depends on such traffic, news sites and blogs have made it easy for their readers to share articles they find interesting, from the ubiquitous share buttons alongside news content, bearing the logos of Facebook, Twitter and others, to so-called "social reader" apps built directly into Facebook. *The Guardian* newspaper, for example, recently announced that, for the first time, more visits to their site were coming through Facebook than through Google search.[1] This user behavior gives us an unprecedented chance to study the readers of news articles at a large scale by analyzing their public digital footprint.

In the past, work has been done that uses such data in the setting of personalized news, recommending articles to a reader based on previous articles that he or she may have shared or liked (e.g., [De Francisci Morales et al., 2012, El-Arini et al., 2009, Li et al., 2010]). This was the setting of Chapter 3 of this thesis. However, in this chapter, we seek to investigate a different question: rather than modeling a reader by the articles he shares, what can we instead learn *about an article* from the attributes of its readers?

In particular, we study this question from two angles:

- Can we build a valuable, general purpose document representation by representing new articles—never before seen or shared—with the predicted attributes of their likely readers?
- Can we produce useful descriptions of writers, news sources, politicians, pundits and other public figures by analyzing the readers of the articles they publish?

In order to address these questions, we once again utilize the microblogging site Twitter[2] as a testbed, as it is widely used by readers as a public medium for disseminating articles and other interesting links. In particular, Twitter users share articles by *tweeting* them (Figure 6.1). Moreover, as described in the previous chapter, many Twitter users also describe themselves in a short profile description, using words like "vegetarian" or "runner" (Figure 6.2). Following the convention of Chapter 5, we will refer to these user attribute labels as *badges*. As most Twitter profiles are public, we can thus scan millions of tweets to learn the relationship between articles and the badges of users who share them. In contrast to Chapter 5, we are not interested in predicting badges of individual Twitter users, but rather are interested in representing arbitrary documents in terms of badges learned from Twitter.

To look at an article through the lens of its readers, one could directly analyze the profiles of all the Twitter users who have shared the article. This approach, however, is impossible to extend to articles not shared extensively on Twitter. We thus take the more general approach of associating badges with the *content of the articles* rather than directly with the articles. More specifically, we learn a sparse, labeled topic model over all the articles tweeted in our Twitter testbed, associating each topic with a badge. For example, if users who have the word "vegetarian" in their Twitter profiles often share articles about health food, then

[1] `http://www.guardian.co.uk/gnm-press-office/changing-media-summit-tanya-cordrey`
[2] `http://www.twitter.com`

Figure 6.1: Example of a tweet that includes a link. We focus on tweets which link to news articles.

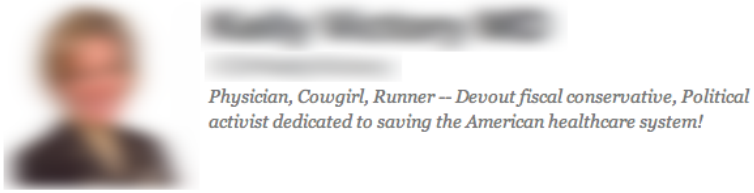


Figure 6.2: Example of a Twitter user profile. Badges such as "physician," "fiscal," or "healthcare" all describe the user's interests.

we might learn a "vegetarian" topic with high weights on the words "tofu" and "kale." We treat these topics as a labeled dictionary, which we can then use to represent new articles.

Figure 6.3 shows an example article from *The Guardian*, "Haqqani network is considered most ruthless branch of Afghan insurgency,"[3] represented as both words and badges. This article is about a particular militant group operating on the Afghanistan-Pakistan border, and in Figure 6.3a, we see that the most important words in this document correspond to the name of this network: the *Haqqani group*. While informative, such a representation of the article could be too fine-grained, in that a related article about a different aspect of the conflict in Afghanistan might not score high on a similarity score, if it does not specifically talk about the Haqqani network. An alternative representation might act at a higher level of abstraction, as we see in Figure 6.3b, which is the badge-based representation of this article which we develop in this chapter.

Modeling article content through user attributes, in comparison with user-oblivious approaches such as latent Dirichlet allocation (LDA) [Blei et al., 2003], offers a more interpretable and natural representation of the articles for personalization and recommendation algorithms. For example, if we are to represent a user's preferences as a weight vector over the features of a document collection, as is commonly done in content-based filtering [El-Arini et al., 2009, Yue and Guestrin, 2011], we expect that a user is more likely to have a coherent preference for, say, a "vegetarian" badge than an arbitrary topic from a topic model.

Another advantage of using badges to represent articles is that by associating the relatively stationary badges with the highly dynamic latent topics, one can naturally match the corresponding latent topics across different time periods. For instance, while what it means to be "liberal" changes from month to month, as expressed in what self-described liberals share on Twitter, the "liberal" badge is persistent, and allows us to immediately produce a correspondence between"liberal" topics trained on tweets from different periods of time. In contrast, in a traditional topic modeling setting, without using user attributes, we would be

[3] http://www.guardian.co.uk/world/2012/sep/07/haqqani-network-blacklisted-terrorist-us

(a) words

(b) badges

Figure 6.3: In this figure, we see the difference between the word representation and badge representation of the same article from *The Guardian*, "Haqqani network is considered most ruthless branch of Afghan insurgency" (September 7, 2012). The badge representation is based on attributes of the likely readers of the article, and exists at a higher level of abstraction than the word representation. In (a), the size of a word is proportional to its tf-idf weight, while in (b), the size of a badge is proportional to the weight it is assigned via the approach in Section 6.3.2. We continue the convention of the previous chapter, and display badges in blue and words in black.

forced either to perform a heuristic bipartite matching on the topic-word distributions from the different time periods, to best match the unlabeled topics with each other, or to resort to a more complicated topic model that directly models the time stamp of each article, which can lead to inefficient inference [Blei and Lafferty, 2006].

We perform extensive evaluation of our approach, and show through both examples and quantitative experiments that incorporating reader information into content analysis yields article representations that are more interpretable for human understanding and more effective for various practical applications.

6.2 Approach Summary

We give a succinct high-level summary of our model and algorithms in this section and provide full details in the following sections of the chapter:

1. We collect a training data set of tweeted news articles from a specified time period. We represent the content of each training article as a bag-of-words vector, with more important words accruing larger weight.

2. We learn a labeled dictionary—whose columns correspond to badges and rows correspond to words in a vocabulary—by minimizing the (regularized) reconstruction error of training articles with respect to the badges of the users who shared them.

3. Given a new article from the same time period, we represent the article as the sparse linear combination of badges that most faithfully represents the article's content in terms of the labeled dictionary learned in the previous step. We enforce that the set of badges selected in this step are related through a structured sparsity regularization.

6.3 The Badge Model

The data we gather from Twitter is threefold: (1) We take each tweeted article, download its content, and represent it as a vector of words following the bag-of-words convention; (2) We associate each article with the users who have tweeted it; (3) We associate each user with a set of descriptive words from his or her profile, which we refer to as *badges*.

Given this data, we can then describe each badge by a small weighted set of characteristic words and describe an article by a small weighted set of the badges, selecting badges whose characteristic words collectively best represent the content of the article.

We emphasize that we want sparsity in two parts of our model: we would like each badge to have a small set of characteristic words and we would like each article to be described by a small set of badges. These sparsity assumptions reduce model complexity and conform with assumptions on natural human cognition, leading to models that are much faster to compute and much easier to interpret.

Formally, we let V denote the size of the vocabulary in our training data, N the number of training documents, and K the total number of badges. From a generative perspective, we think of the document i, represented as a V-dimensional vector of the words, $\mathbf{y}_i$, as formed by:

$$\mathbf{y}_i \approx \mathbf{B}\boldsymbol{\theta}_i. \tag{6.1}$$

$\mathbf{B}$ is a non-negative $V \times K$ matrix with a column for each badge, representing the weighted set of characteristic words for the badge. $\boldsymbol{\theta}_i$ is a K-dimensional vector that similarly represents the weighted set of characteristic badges associated with document i. We borrow a term from information theory and refer to $\mathbf{B}$ as our *badge dictionary* where each column of $\mathbf{B}$ is an entry in the dictionary. Our sparsity assumptions translated in this setting means that $\mathbf{B}$ and $\boldsymbol{\theta}_i$ must both have small numbers of non-zero entries. The training corpus of articles along with the user profile information provides us the $\mathbf{y}_i$'s and the approximate $\boldsymbol{\theta}_i$'s for many documents with which we can learn the matrix $\mathbf{B}$; we refer to this phase as "learning the dictionary." We then can apply the dictionary $\mathbf{B}$ to analyze contents of new documents and estimate the corresponding relevant badges for the new documents by estimating the $\boldsymbol{\theta}_i$ vector; we refer to this phase as "coding the documents."

6.3.1 Learning the Dictionary

For each document i in our training corpus, we observe the content vector $\mathbf{y}_i$, as well as the badges of the readers who shared document i on Twitter. The set of all badges of the readers does not give us direct access to $\boldsymbol{\theta}_i$ because it may contain badges irrelevant to the document as well as omitting badges important to the document. As the previous chapter showed, there is a *probabilistic* relationship between the badge labels present in a user's profile and the user's true set of badges. However, here, we are not interested in the specific badges of a particular user, but rather a higher-level association between badges and the articles shared by a large collection of users. Thus, it is reasonable and sufficient to say that, with a large number of users and documents, the documents shared by readers self-identified with a specific badge k will be on average relevant to the badge k and the documents shared by readers not self-identified with a badge k will be on average irrelevant to badge k. Therefore, in order to learn the dictionary $\mathbf{B}$, we can approximate $\boldsymbol{\theta}_i$ by taking each of the readers of document i, and assume a uniform distribution over the badges each of them declares in his or her profile. We then estimate $\boldsymbol{\theta}_i$ by aggregating over document i's readers.

For example, we consider an article i that was shared on Twitter by two users:

- Alice's Twitter profile contains the badges "liberal" and "feminist";
- Bob's Twitter profile contains the badges "liberal," "football" and "German."

In this case, we would assume Alice is half-liberal and half-feminist, while Bob is one third each: liberal, football and German. We would thus estimate $\boldsymbol{\theta}_i$ as the point-wise average of the two vectors: $\langle \text{liberal} : 0.5,\ \text{feminist} : 0.5 \rangle$ and $\langle \text{liberal} : 1/3,\ \text{football} : 1/3,\ \text{german} : 1/3 \rangle$. Formally:

$$\theta_{ik} \approx \sum_{u \in\ Tweeted_i} \frac{\lambda_k^{(u)}}{\sum_\ell \lambda_\ell^{(u)}}, \tag{6.2}$$

where we continue the notation of the previous chapter and let $\lambda_k^{(u)} = 1$ if and only if user u identifies himself with badge k in his Twitter profile.

With each $\mathbf{y}_i$ given and each $\boldsymbol{\theta}_i$ approximated, the badge dictionary $\mathbf{B}$ can be learned by choosing a loss function and minimizing the loss objective:

$$\min_{\mathbf{B} \geq 0} \sum_{i=1}^{N} l(\mathbf{y}_i, \mathbf{B}\boldsymbol{\theta}_i) + \lambda_B \sum_{j=1}^{V} \sum_{k=1}^{K} |\mathbf{B}_{jk}|. \tag{6.3}$$

Note that we also constrain all entries of $\mathbf{B}$ to be non-negative to make the results more interpretable and use the well-studied ℓ_1-regularization on the entries of $\mathbf{B}$ to encourage sparsity in the learned $\mathbf{B}$ matrix.

In this chapter, we let $\mathbf{y}_i$ be a *term frequency-inverse document frequency (tf-idf)* vector of the words in document i, normalized to have ℓ_2-norm of 1.[4] The $\boldsymbol{\theta}_i$ vector, as described before, gives a uniform weight to the set of all badges of the readers, normalized also to have unit ℓ_2-norm. We then minimize a square-error loss and choose the regularization parameter λ_B that achieves a desired level of sparsity in the resulting $\mathbf{B}$ matrix:

$$\min_{\mathbf{B} \geq 0} \sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{B}\boldsymbol{\theta}_i\|_2^2 + \lambda_B \sum_{j=1}^{V} \sum_{k=1}^{K} |\mathbf{B}_{jk}|. \tag{6.4}$$

We optimize Eq. (6.4) using a simple projected stochastic gradient descent, as outlined in Algorithm 6.1.[5] This approach to optimization allows us to operate on large, streaming, Web-scale data sets.

It is interesting to consider the alternative objective function where we let $\mathbf{y}_i$ and the columns of $\mathbf{B}$ represent probability distributions over words and let $\boldsymbol{\theta}_i$ represent a probability distribution over the set of badges.

[4] A tf-idf vector representation of a document is an element-wise product of two vectors, each with dimension equal to the vocabulary size: **(1)** the *term frequency* (tf) vector stores a count of each unique word in the document; and, **(2)** the *inverse document frequency* (idf) vector stores the log inverse of the proportion of total documents (in a reference or training corpus) containing each word. Specifically, the jth index of a tf-idf vector for document i in a corpus of documents $\mathcal{D}$ can be computed as follows:

$$\text{tf} - \text{idf}_i(j) = \#(j,i) \cdot \log\left(\frac{|\mathcal{D}|}{|\{d \in \mathcal{D} : \#(j,d) > 0\}|}\right),$$

where $\#(j,i)$ indicates the number of times word j occurs in document i. The tf-idf representation is widely used in information retrieval settings, as it is a simple yet effective way of down-weighting words with little discriminative value (i.e., common words that appear uniformly across a document collection).

[5] Further details on initialization and parameters can be found in the appendix to this chapter, Appendix 6.8.

Algorithm 6.1: Projected stochastic gradient descent for learning the badge dictionary $\mathbf{B}$

// Data:
$\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N]$
$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]$
// Parameters:
Let m be the minibatch size
Let ϵ be the termination tolerance
Let α_0 be an initial step size parameter
Let λ_B be the sparsity regularization parameter
// Initialization:
$\mathbf{B}^{(0)} = \infty$
Set $\mathbf{B}^{(1)}$ to an initial value
$t \leftarrow 1$
// Termination condition
while $\|\mathbf{B}^{(t)} - \mathbf{B}^{(t-1)}\|_F > \epsilon$ **do**
  // Random minibatch
  $\mathcal{M} \leftarrow \text{randsample}(N, m)$
  $\widehat{\boldsymbol{\Theta}} \leftarrow \boldsymbol{\Theta}_{:,\mathcal{M}}$
  $\widehat{\mathbf{Y}} \leftarrow \mathbf{Y}_{:,\mathcal{M}}$
  // Compute gradient
  $\nabla \leftarrow -\frac{1}{m}\left(\widehat{\mathbf{Y}} - \mathbf{B}^{(t)}\widehat{\boldsymbol{\Theta}}\right)\widehat{\boldsymbol{\Theta}}^{\mathsf{T}}$
  // Set step size
  $\alpha = 1/\sqrt{\max(\alpha_0, t)}$
  // Take gradient step
  $\mathbf{B}^{(t+1)} \leftarrow \mathbf{B}^{(t)} - \alpha\nabla$
  // ℓ_1-projection via soft-thresholding, while maintaining non-negativity
  $\mathbf{B}^{(t+1)}_{>0} \leftarrow \mathbf{B}^{(t+1)}_{>0} - \lambda_B$
  $\mathbf{B}^{(t+1)}_{<0} \leftarrow 0$
  $t \leftarrow t + 1$

We can then minimize the information loss, also known as the Kullback-Liebler divergence:

$$\min_{\mathbf{B}\in\Delta} -\sum_{i=1}^{N} \mathbf{y}_i^{\mathsf{T}} \log(\mathbf{B}\boldsymbol{\theta}_i), \tag{6.5}$$

where the notation $\mathbf{B} \in \Delta$ constrains all columns of $\mathbf{B}$ to lie in the probability simplex, i.e., be non-negative and sum to one. It is important to note that we do not include the ℓ_1-regularization in this approach because the probability simplex constraint already requires that $\sum_{j=1}^{V} |\mathbf{B}_{jk}| = 1$ for all solutions, thus rendering the regularization irrelevant. Interestingly, this optimization is exactly equivalent to maximum likelihood inference in the probabilistic latent semantic indexing (pLSI) model, assuming fixed topic weights $\boldsymbol{\theta}_i$ [Hofmann, 1999]. In this chapter, however, we choose to work with the square-error loss primarily because we get more direct control over the sparsity level of the estimators.

Many techniques learn both $\mathbf{B}$ and $\boldsymbol{\theta}_i$ from the training corpus—non-negative matrix factorization and LDA, for example. However, joint estimation of both $\mathbf{B}$ and $\boldsymbol{\theta}_i$ is inherently a much more complex problem than learning just one; the resulting joint learning algorithms are not only slower, but require the training

corpus to be both larger and more regular. Thus, one way to understand our approach is that we use the reader attribute information to drastically reduce the complexity of many content analysis models.

6.3.2 Coding the Documents

A straightforward approach for representing a new document in terms of the badges is to take the same loss-objective as in the dictionary learning phase, and optimize over $\boldsymbol{\theta}_i$ instead of $\mathbf{B}$. That is, given a new document i, we optimize the following:

$$\min_{\boldsymbol{\theta}_i \geq 0} l(\mathbf{y}_i, \mathbf{B}\boldsymbol{\theta}_i) + \lambda_\theta \|\boldsymbol{\theta}_i\|_1, \tag{6.6}$$

where we again encourage sparsity in the estimated $\boldsymbol{\theta}_i$ by the ℓ_1-regularization.

With squared-error, our objective takes the form of the well-known non-negative lasso:

$$\min_{\boldsymbol{\theta}_i \geq 0} \|\mathbf{y}_i - \mathbf{B}\boldsymbol{\theta}_i\|_2^2 + \lambda_\theta \|\boldsymbol{\theta}_i\|_1. \tag{6.7}$$

We once again borrow a term from information theory and refer to an optimization like in Eq. (6.7) as *coding the article* in terms of the badges. Eq. (6.7) can be solved efficiently through various algorithms, including coordinate descent and Shotgun [Bradley et al., 2011].

6.3.3 Incorporating Relations among Badges

In practice, there is a subtle problem with the formulation in Eq. (6.7). Many badges tend to be highly related, such as "progressive" and "liberal," "school" and "student," and "vegan" and "vegetarian." These closely-related badges tend to model similar content and overlap in explanatory power. Thus, the estimated set of relevant badges—the non-zero entries of the estimated $\boldsymbol{\theta}_i$ vector, encouraged to be as small as possible by the sparsity regularization—would arbitrarily include, e.g., either "progressive" or "liberal," but not both. The fact that these choices are arbitrary has undesirable consequences: for instance, given two very similar articles about the liberal political view on education, one may be represented by the badges "progressive" and "school" and the other by a completely disjoint set of badges, "liberal" and "student". Any learning algorithm that uses the selected badges as features would consequently be misled into treating the two articles as completely dissimilar.

Ameliorating this problem requires two steps: (1) we must first detect similarity relations among the badges; and, (2) we must then augment the article coding objective so that groups of closely related badges would be selected together in the article representation.

To determine whether two badges are related, we look at co-occurrence counts of the badges in the profiles of Twitter users. Closely related badges would either frequently co-occur in users' profiles—in cases like "Obama" and "liberal"—or frequently co-occur with some other common badges—in cases like "liberal" and "progressive"—with the common badge perhaps being, e.g., "activist" or "blogger." To take into account both cases, we form a weighted undirected graph over the badges where each edge between two badges has a weight proportional to the frequency that these two badges co-occur in Twitter user profiles. More precisely, if s and t represent two distinct badges, we let the weight of the edge between s and t be $w_{st} \equiv \frac{\#s, t \text{ co-occur}}{(\#s \text{ occur})(\#t \text{ occur})}$. One can see then that highly related badges would either be neighbors in this

(a) without graph regularization (b) with graph regularization

Figure 6.4: The effect of graph regularization on an article about Mac OS X Lion. The size of a badge is proportional to its weight.

graph or be connected by a very short path, where the weights of the edges on the path would be very high.

For the second step, we augment our model with the so-called *graph-guided fused lasso* regularization of Kim et al. [2009]:

$$\min_{\boldsymbol{\theta}_i \geq 0} \|\mathbf{y}_i - \mathbf{B}\boldsymbol{\theta}_i\|_2^2 + \lambda_\theta \|\boldsymbol{\theta}_i\|_1 + \lambda_\mathcal{G} \sum_{(s,t)\in E(\mathcal{G})} w_{st} |\boldsymbol{\theta}_i(s) - \boldsymbol{\theta}_i(t)|, \tag{6.8}$$

where w_{st} is the co-occurrence weight of the badge pair (s,t) in the co-occurrence graph $\mathcal{G}$, as defined above. The graph fusion regularization encourages $\theta_i(s)$ to be close to $\theta_i(t)$ for all edges (s,t) in the graph where the strength of the regularization is proportional to the weight of the edge. In this way, highly related badges, being closely connected in the co-occurrence graph by heavily weighted edges, would be incentivized to simultaneously have either all zero or all non-zero values in entries of $\boldsymbol{\theta}_i$. The graph fusion regularization parameter, $\lambda_\mathcal{G}$, regulates how big a role the graph $\mathcal{G}$ should play in regularizing $\boldsymbol{\theta}$. We refer the readers to the recent work of Chen et al. [2012] for a detailed discussion of the optimization algorithm for solving Eq. (6.8), which we use in our approach.

As an example of how this graph regularization addresses our problem, we can consider an article about Mac OS X Lion.[6] Coding this article with the vanilla lasso, without graph regularization, leads to a badge representation overwhelmed by the "lions" badge. This is problematic because, while the "lions" badge well explains the word "lion," which appears often in the article, the main usage of the "lions" badge occurs in the context of the Detroit Lions football team. As a result, the Mac OS X article could, with respect to the computed badge representation, be more similar to a football article than to a technology article. When using the graph-guided fused lasso however, we obtain a more balanced coding, with the badge "apple" and "geek" now being the most dominant, taking up nearly sixty percent of the squared two-norm of the badge vector (cf. Figure 6.4).

The reason for this improvement is evident when we consider the neighbors of"lions" and "apple" in our badge graph. The strongest links emanating from the"lions" badge are related to Michigan—e.g., "Detroit" and "mlive" (a Michigan news site)—or to animals—e.g., "jungle," "monkey" and "roar." These neighboring badges do not do a good job explaining the Mac OS X article, and so this forces "lions" to be downweighted. However, if we consider the strongest neighbors of "apple" in the badge graph, we see words such as "fanboy," "jailbreak" and "ipod," which are much more related to the content of the article.

[6] http://www.macobserver.com/tmo/article/my_favorite_stealthy_lion_features/

Table 6.1: Training data statistics

Time period	Tweeted links with user profiles	Tweeted news articles in English	Vocabulary size	Number of badges
Sep. 2010	18,872,925	596,522	51,182	4,460
Sep. 2011	38,158,817	847,077	55,688	5,029
Sep. 2012	67,346,626	1,514,670	58,235	5,247

6.4 Experimental Results

We conduct an extensive empirical analysis of our badge-based concept representation, focusing on the questions we posed at the beginning of the chapter. Specifically, we seek to show that by representing documents by attributes of their likely readers, we can create a document representation suitable for personalization—particularly as part of our interactive concept coverage framework—while simultaneously gaining interesting insights into the works of writers, politicians and other public figures.

We begin by describing the large data set we use for our evaluation, followed by both anecdotal descriptions and quantitative comparisons, showing that our badge-based concept representation is useful and insightful.

6.4.1 Data Processing and Experimental Setup

In order to evaluate our method, we must obtain a training set of tweeted news articles. We achieve this with access to the Twitter Garden Hose stream, which is an approximately 10% random sample of all tweets.[7] In our reported experiments, we consider three months-worth of tweets: September 2010, September 2011 and September 2012.[8] For each of these three months, we scan through every tweet in the Garden Hose and extract those that are: (1) a tweet of a link; and, (2) came from a user with a non-empty profile. This leaves us with over 120 million tweets across the three months (cf. Table 6.1).

Next, as we are particularly interested in news articles, and not videos, photos, games and other such shared web pages, we filter the tweeted links to match one of 20,000 mainstream news sources, as defined by Google News.[9] We then *download each news article* shared in this set of tweets that we believe to be written in the English language, resulting in a smaller, but extremely rich, data set, consisting of nearly 3 million tweeted news articles.

We use standard heuristics to extract the most meaningful unique words in these articles to create a vocabulary for each time period, as well as extract all badges that occur more frequently than a specified threshold. Statistics can be found in Table 6.1.

Based on this training data, for each of the three months, we can compute the $\mathbf{\Theta}$ and $\mathbf{Y}$ matrices, as well as the undirected graph over badges with weights w_{st}, and commence with dictionary learning, as described above in Section 6.3.1. We learn a separate badge dictionary for each of the three months.

[7] We are grateful to Brendan O'Connor and Noah Smith for providing us with this access.

[8] It is important to note that, throughout the development of our approach and algorithms, we used a held-out validation set of tweets and tweeted articles, corresponding to July 2011 and July 2012.

[9] List of Google News news sources provided by Jure Leskovec, to whom we are grateful.

Table 6.2: Number of articles per section of *The Guardian* in our test set

Time period	World	Sport	Opinion	Business	Life & Style	Science	Technology	UK
Sep. 2010	942	980	636	614	668	184	299	134
Sep. 2011	1,198	1,146	621	627	547	201	335	217
Sep. 2012	1,162	1,044	603	466	437	143	259	203

For the quantitative comparisons, we require a test set of articles. While our training requires the analysis of tweets, any documents—including never-before-published ones—can be represented using our badge-based concept representation. Thus, for our test set, we download eight entire sections from *The Guardian*, a leading British newspaper, over the three months considered in our training set, comprising nearly 14,000 articles. We represent each test article as a tf-idf vector over the time-specific vocabulary constructed during training. We can then code each article by optimizing Eq. (6.8), using the dictionaries learned from the training data. Statistics of the test set can be found in Table 6.2.

More details on the entire data processing pipeline can be found in Appendix 6.7, while information on parameter settings for our optimization can be found in Appendix 6.8.

6.4.2 Examples

After learning our badge dictionary on the training set, spanning three months, we can ascertain how well the badge-labeled topics capture semantic themes in our data.

Top Ten At first glance, we can examine the ten badges that we use the most (i.e., highest total weight) to code the *Guardian* articles from September 2012, in our test set:

1. Guardian
2. Olympics
3. London
4. cricket
5. soccer
6. premier (as in, English Premier League)
7. tennis
8. fashion
9. gossip
10. Labour (as in, the British political party)

As we see in Figure 6.5, the words representing these ten badges align quite well with what we would expect. The most prevalent badge—"Guardian"—acts as a "background" badge in this particular data set, while the next six badges describe different aspects of sports, which represents a large proportion of our data set (cf. Table 6.2). The first badge that we would consider to be mediocre is the "views" badge, down at the 27th spot in the ranking. The characteristic words of this badge seem to be unrelated to the concept

(a) 1. Guardian (b) 2. Olympics (c) 3. London

(d) 4. cricket (e) 5. soccer (f) 6. premier

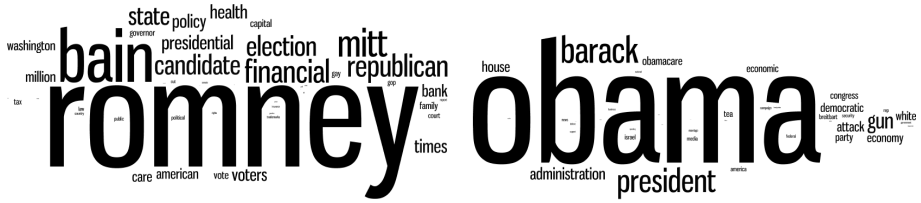
(g) 7. tennis (h) 8. fashion (i) 9. gossip

(j) 10. Labour (k) 27. views

Figure 6.5: **(a)-(j)** The top ten badges used to code articles from *The Guardian* in our September 2012 test data set. The size of a word is proportional to its weight in the badge. **(k)** This word cloud represents "views," the 27th most heavily used badge when we code the September 2012 *Guardian* articles. This is the first badge in the ranking whose word representation is not what one would expect.

"views," giving us an example where our dictionary learning phase produced a poor dictionary element. However, we find it heartening that the majority of the badges we learn are of high quality, as evident in Figure 6.5.

Dueling Badges An interesting exercise is to take a pair of related badges, and see how their word representations compare. In Figure 6.6, we see a comparison of two popular badges related to American politics: "progressive" (a popular liberal badge) and "tcot" (**T**op **C**onservatives **o**n **T**witter). These dictionary elements were learned from the 2012 data, and thus come from the heat of the American Presidential race between Barack Obama and Mitt Romney. As this race was heavy on negative campaigning (cf., for example, [Greenstein, July 17, 2012]), it is not surprising to see that progressive supporters of



(a) progressive − tcot (b) tcot − progressive

Figure 6.6: Here, we see the relationship between two related badges: "progressive" and "tcot" (Top Conservatives on Twitter). The word cloud on the left contains words that are more important for "progressive" than for "tcot," with the size of the word proportional to the difference in weights between the two dictionary elements. On the right, we see a word cloud containing the converse: words that are more important for "tcot" than for "progressive."

(a) music (2010) (b) music (2012)

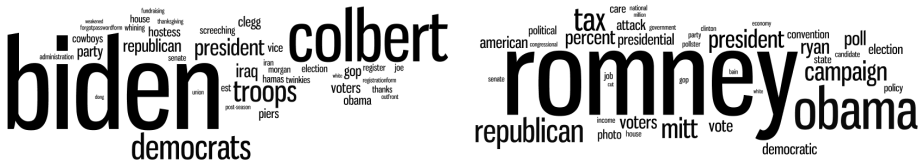
Figure 6.7: The "music" badge is one of the most static badges in our data set; its characteristic words barely change over the two year period from September 2010 to September 2012, as can be seen in this pair of word clouds. Again, the size of a word is proportional to its weight in the dictionary element.

Barack Obama were more likely than conservatives to share articles about Mitt Romney, and in particular, his controversial ties to Bain Capital, a financial firm he once headed. Likewise, conservatives are more likely than progressives to share articles about Barack Obama, presumably critical of him.

Badges Over Time One motivation for using badges to represent documents is their persistence over time. For example, even if what it means to be liberal changes from month to month and year to year, the "liberal" badge is always there to represent liberal-leaning documents. Thus, it is instructive to consider examples of both static and dynamic badges.

In Figure 6.7, we find the "music" badge, which is one of the most static badges in our data set; its characteristic words barely change over the two year period from September 2010 to September 2012. Namely, the type of Twitter user who identifies herself with music in her profile is likely to share articles with the words "music," "band," "album" and "song."

In contrast, Figure 6.8 shows one of the most dynamic badges in our data set: the one representing Vice President Joe Biden. The type of user who identifies himself with "Biden" shares rather different articles in 2010 and 2012. In September 2010, such a user focuses on the Vice President as well as comedian Stephen Colbert, who at the time was co-hosting a political rally with fellow comedian Jon Stewart. However, by 2012, all signs of Joe Biden have diminished, and the primary focus of this badge is on the American Presidential race.



(a) Biden (2010) (b) Biden (2012)

Figure 6.8: The "Biden" badge is a dynamic one. In 2010, readers with the badge share articles about Joe Biden and Stephen Colbert, while in 2012, the focus turns to Barack Obama and Mitt Romney, due to the Presidential campaign.

6.4.3 Case Study with Political Columnists

To demonstrate how our badge representation can provide insight on the makeup of a writer's likely readers, we use our model to analyze fourteen notable political columnists in the United States. These columnists each specialize in different topics, from economics to foreign policy, and are perceived to have different political leanings from very liberal to ultra-conservative. We show through various examples that, by understanding the writings of these political columnists through badges, we can characterize their target audiences in interesting ways. We emphasize that we look at only the content of the columnists' articles; only the badge dictionary is learned from documents shared on Twitter, and thus this analysis does not require that the columnists' articles appear on Twitter at all.

As a first analysis, we take each article written by each of the fourteen columnists in July 2012, and code the article text in terms of badges, using our methodology. For each columnist, we then average the badge representations of the columnist's articles, resulting in an aggregate badge representation for each columnist. Examples can be found in Figure 6.9. We find that the badge representation, in almost all cases, accurately reflects the topics of expertise of the columnists; for instance, the words "aid" and "Africa" appear prominently in the badge representation for Nicholas Kristof, which makes sense because a reader who is self-described to be interested in "aid" or "Africa" would be quite likely to read Kristof's analyses of the various humanitarian crises in third world countries. Likewise, the badge representation for Maureen Dowd accurately shows that her likely readers are "progressive." It is critical to point out that Dowd does not in fact use the word "progressive" in any of her columns throughout this time period; rather, this coding corresponds to the attributes of her likely readers. Additionally, the badges "Irish" and "Ireland" appear prominently because Maureen Dowd was on assignment in Ireland in July 2012, writing prolifically about the country.

As a second analysis, we compare the political leanings of the likely readers of the fourteen columnists, by coding the columnists' articles in terms of *only* the "progressive" and "tcot" badges. In Figure 6.10, we place the columnists on a spectrum, where the location of each columnist is based on the relative weight of the "tcot" badge to the "progressive" badge in his or her average badge representation. Thus, columnists appearing on the left side of the spectrum are more likely to appeal to readers self-identified as "progressive" than to readers self-identified as "tcot." For example, the location of ultra-conservative writer Ann Coulter on the far right of the spectrum indicates that, based on her writings in July 2012, her likely readership during that month is almost exclusively conservative.

Overall, our ranking of the political columnists roughly line up with the public perception of the columnist' political alignments, with outspoken conservative voices like Coulter and Charles Krauthammer placed

(a) Nicholas Kristof

(b) Maureen Dowd

(c) Thomas Friedman

(d) Paul Krugman

Figure 6.9: These four word clouds depict the prominent badges that best represent the writings of four well-known *New York Times* columnists. The badge representations of the writers match well with the subject matter of their columns. Nicholas Kristof (a) writes extensively about poverty and social justice abroad; Maureen Dowd (b) is a liberal columnist who writes general political commentary; Thomas Friedman (c) specializes in foreign policy, and in particular, the Middle East; Paul Krugman (d) is a Nobel laureate economist who advocates left-leaning economic policy in his columns.

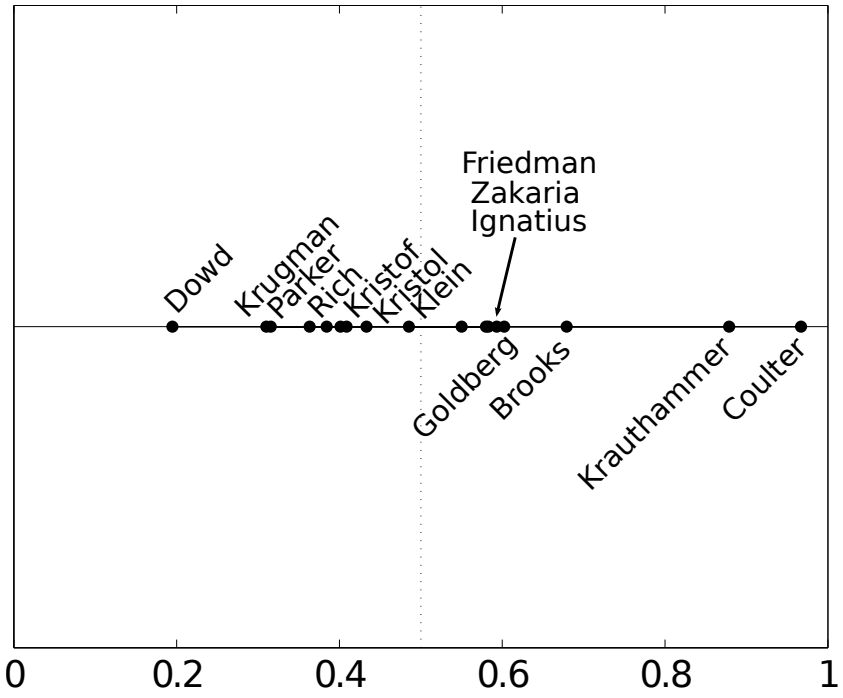


Figure 6.10: Predicted political alignments of the likely readers of fourteen political columnists based on results from representing the columnists' writing with only two badges: "progressive" and "tcot." The columnists are ranked from left to right based on how well the content of their writing is captured by the left-wing "progressive" badge as opposed to the right-wing "tcot" badge.

on the far right, and well known liberal voices like Maureen Dowd on the (not as extreme) left. Likewise, Thomas Friedman, Fareed Zakaria and David Ignatius are clustered together, as they all write primarily about foreign policy. It is important to emphasize, however, that our approach does not attempt to directly classify the political alignment of the columnists. Rather, we instead try to identify what kind of readers would be interested in reading each of the columnist's editorials. For instance, if we take the example of Kathleen Parker, although she is a conservative columnist with the *Washington Post*, she often levels strong criticism against the Republican Party, and even supported Barack Obama for the 2008 presidential elections. It is thus sensible that she is placed more to the left in our chart despite being conservative, because politically liberal readers often enjoy reading her columns.

However, our ranking is not perfect. If we consider the location of William Kristol, a neo-conservative icon and founder of *The Weekly Standard*, we find that he is incorrectly placed on the left side of this spectrum. We hypothesize that this behavior arises from the phenomenon visualized in Figure 6.6, whereby progressives are more likely to write about Mitt Romney than conservatives. While an unabashed conservative, in the month of July 2012, Kristol writes about Romney nearly twice as much as he writes about Obama, which may explain the discrepancy.

Finally, it is interesting to note the relationship between our case study and the well-studied *ideal point model* from quantitative political science, which assumes legislators and bills lie in a low-dimensional Euclidean space indicating political positions, and the affinity of a legislator for a bill is a function of the distance between their two locations [Poole and Rosenthal, 1985]. Early work learned such ideal points from roll call votes, whereas more recent work in machine learning has combined roll call data with text analysis [Gerrish and Blei, 2011]. While perhaps visually similar, Figure 6.10 is computed based on a completely different signal than traditional ideal point models.

A full listing of the fourteen columnists and the articles we use in this analysis appears in Appendix 6.9.

6.4.4 Quantitative Comparisons

In this section, we consider three quantitative analyses, evaluating the performance of our badge-based concepts as a document representation. We compare our approach to two commonly used concept representations: a fine-grained tf-idf representation and a coarse-grained 100 topic LDA-based representation. These are the same two representations we evaluate in Chapter 3 of this thesis.

Coherence As a first comparison, we test our hypothesis that the concepts we learn with our badge-based representation are more semantically coherent than topics learned through a topic model, such as LDA. Our belief stems from the intuition that there is no explicit incentive for a topic model to produce coherent topics corresponding to human interests, while each of our dictionary elements corresponds to a badge directly used by a Twitter user to describe himself.

To quantitatively measure such a notion of coherence, we use the methodology of Mimno et al. [2011], and compute a statistic based on how frequently the top words in each topic co-occur in a reference corpus. Mimno and colleagues show that this statistic strongly correlates with human notions of coherence, as validated by a user study. Figure 6.11 shows that, using Mimno's metric, our learned badge dictionary produces more semantically coherent concepts than LDA, when trained on the full September 2012 training data set.

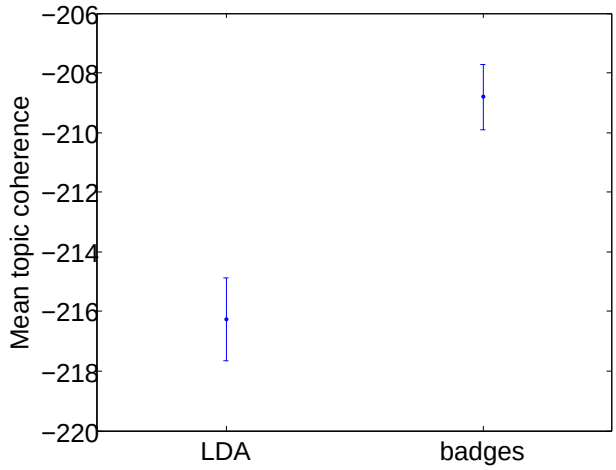


Figure 6.11: Dictionary elements from our learned badge dictionary are more semantically coherent than topics from an LDA topic model. Both models use the same number of topics/badges (5,247), and are trained on the September 2012 training data set. The coherence numbers reported are computed based on the methodology of Mimno et al. [2011].

For a fair comparison, we run LDA with the same number of topics as we have badges for this time period (i.e., 5,247). Moreover, we compute Mimno's coherence statistic using the top 15 words in each topic, following the convention used in their paper. Due to the scale of this problem (over 1.5 million documents, 5,000 topics, 55,000 word vocabulary and 375 million total tokens), we run a distributed implementation of collapsed Gibbs sampling for LDA, over 114 cores, provided as part of GraphLab [Gonzalez et al., 2012, Low et al., 2012].[10]

Odd-one-out Our second quantitative comparison examines our hypothesis that the badge-based concept representation is better suited than competing techniques to represent coherent semantic concepts *over time*. This belief is due to the persistent nature of badges, which allows us to straightforwardly bind together concepts across different time periods. For example, the "football" badge in September 2010 can be directly matched to the "football" badge in September 2012, whereas topics from an unsupervised topic model must be matched in more complicated, less straightforward ways.

To evaluate our hypothesis, we conduct the following intrusion-detection experiment on our *Guardian* test data:

1. Pick two newspaper sections at random from *The Guardian*. Call the first one the *home* section and the second the *intruder* section. For example, we might pick "world" and "sport" as the home and intruder sections, respectively.

2. Pick an article, uniformly at random, from the home section of the September 2010 *Guardian* data. We call this article h_1.

3. Pick an article, uniformly at random, from the home section of the September 2012 *Guardian* data. We call this article h_2.

[10] `http://www.graphlab.org`

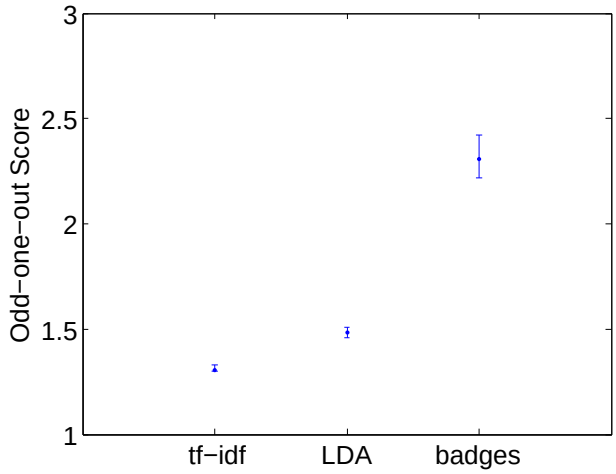


Figure 6.12: Odd-one-out metric showing that our badge-based concept representation does a better job at preserving the semantic similarity of articles within the same newspaper section over time than the competing representations. Results reported are the median of 56,000 independently drawn triplets of articles, and 95% confidence intervals are computed using the normal approximation to the binomial.

4. Pick an article, uniformly at random, from the intruder section of the September 2012 *Guardian* data. We call this article i_2.
5. We compute the (ℓ_2-normalized) concept representations for each of these three articles.
6. For a given concept representation (e.g., for LDA), we compute the following cosine similarity ratio: $(h_1^\mathsf{T} h_2)/(h_1^\mathsf{T} i_2)$. We call this the "odd-one-out" score for this triplet of articles and this concept representation, as it tells us how much more similar the two documents from the same section are to each other, versus the two documents from different sections.

A concept representation with a high "odd-one-out" score indicates that the semantic similarity between articles from the same section is preserved across time. A lower "odd-one-out" score indicates that a concept representation can more easily conflate the content of different news sections, leading to thematic incoherence over time.

We compute this score for our badge-based concept representation, as well as a 100-topic LDA topic representation and a tf-idf word representation. Specifically, for each pair of home and intruder sections, we draw 1,000 random article triplets, and compute the median odd-one-out score for each method. Figure 6.12 shows that, overall, aggregating over all pairs of sections, the badge representation has a significantly better performance on this metric than the two competing techniques.[11] Moreover, if we look at Figure 6.13, we see that this significant advantage holds true not just at an aggregate level, but in about 80% of the individual section pairings. Of the 56 possible pairings, only two resulted in significant wins on this metric by one of the competing techniques.

User Study Our final quantitative evaluation addresses the fundamental question posed at the beginning of this chapter: can we develop a concept representation that works well for personalization?

To answer this question, we conduct a news recommendation user study comparing our badge-based

[11] Confidence intervals about the median are computed using the normal approximation to a binomial (cf. Bland [2000]).

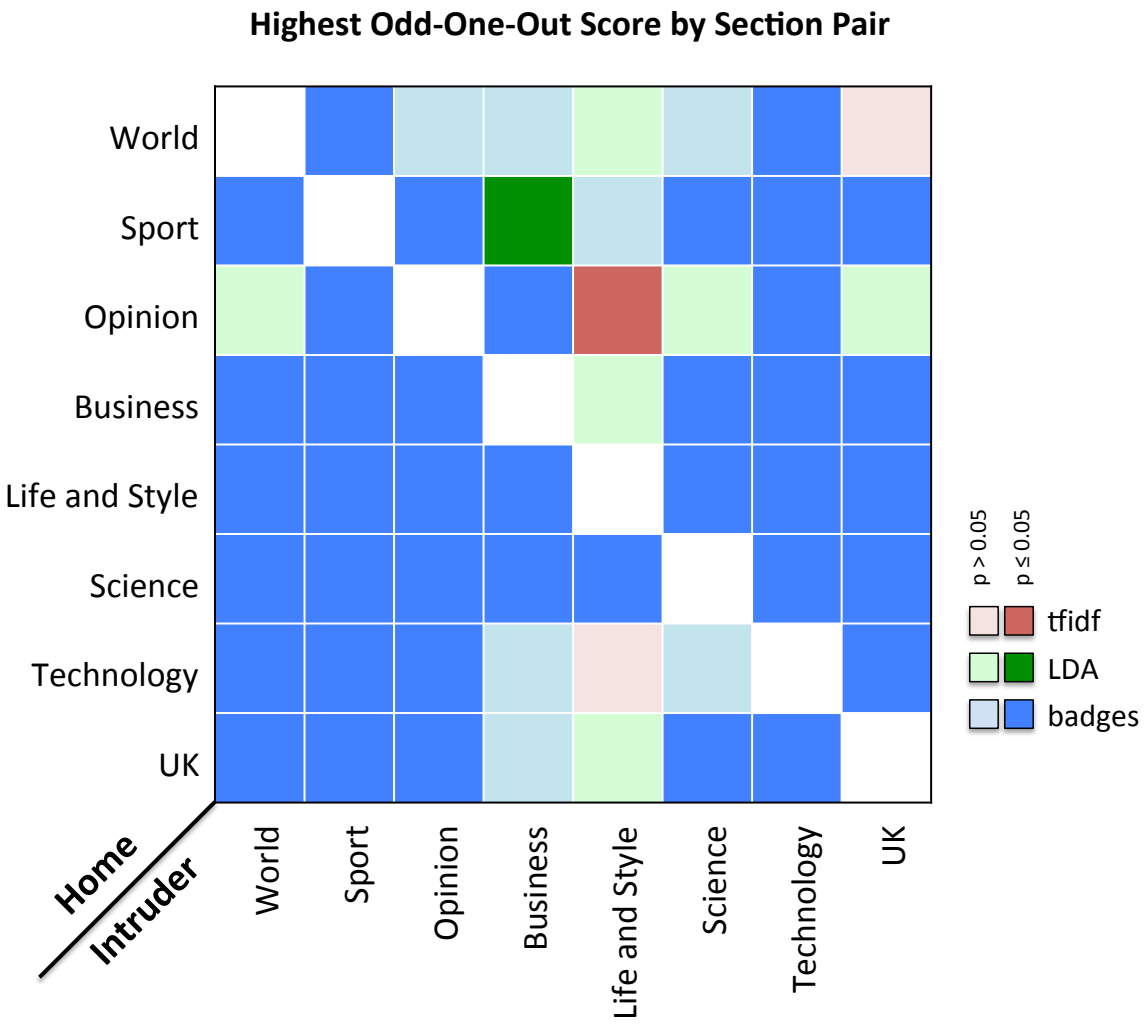


Figure 6.13: In about 80% of potential section pairs from *The Guardian*, the badge-based representation leads to a better odd-one-out score than the competing techniques. Darkly shaded cells are significant at the 95% confidence level, as indicated by the legend.

concept representation to tf-idf and LDA. We use each of the three as concept representations in our interactive concept coverage methodology, introduced in Chapter 1 of this thesis. Our study is in two phases: first, a user provides feedback on a random set of articles that allows us to quickly estimate his interests, and then we recommend articles to the user and measure how many of them he likes.

Specifically, our study involves the following:

1. Pick a random two time periods from the set: { September 2010, September 2011, September 2012 }. Assign one time period to the first phase of the study, and the other time period to the second phase.
2. Draw 20 news articles, uniformly at random, from our *Guardian* test data set, corresponding to the first time period of the study.
3. Present these 20 news articles, one at a time, to the user, asking him to mark each article as interesting or not. This is the first phase of the study.
4. We draw a random concept representation from the set: { tf-idf, LDA, badges }. Based on the concept representation we select, we compute the average concept vector of the articles marked as interesting in the first phase. For example, if the user has indicated interest in just two articles, one on the Manchester United football team and another on the London Olympics, and our randomly selected concept representation was tf-idf, we would average together the tf-idf vectors of the two articles, leading to high weights on words like "London," "football," "Olympics," "Manchester," "premier," "Ferguson," etc.
5. We take the average concept representation computed in the previous step, and convert it so that it matches with the corresponding concepts in the second year of the study. With LDA, this involves the Hungarian algorithm for bipartite matching over the topic-word distributions, while for tf-idf and badges, it involves simply matching the words or labels from one time period to the other. For example, if the time period assigned to the first phase was September 2012, and our second phase corresponds to September 2010, we need to select and permute the indices of the September 2012 average concept vector so that it matches with the ordering of September 2010. Thus, if the word "France" appears in index 3 in 2012, but in index 100 in 2010, the concept weight over "France" needs to move to index 100.
6. We use the transformed average concept vector, indicating the user's interests from phase one of the study, as concept weights for our interactive concept coverage methodology. We employ the greedy submodular maximization algorithm to optimize Eq. (3.3) with a budget of ten, resulting in a diverse set of ten articles from the second time period relevant to the user's interests.
7. We show the recommended articles to the user, one at a time, and obtain feedback on which ones were interesting or not.

We recruited 118 participants for our study on Amazon Mechanical Turk,[12] offering $0.20 per study completion. As our articles are from *The Guardian*, a British newspaper, we require participants that have good English language skills and meaningful ties to the United Kingdom. As we were unsuccessful in recruiting participants directly from Great Britain, we instead recruited exclusively from India. To improve the quality of our test set for our participants, we removed articles that were shorter than 1,200 characters, as well as those that contained the words "rugby" or "cricket."

Figure 6.14 shows that our badge-based concept representation significantly outperforms both tf-idf and

[12] `http://www.mturk.com`

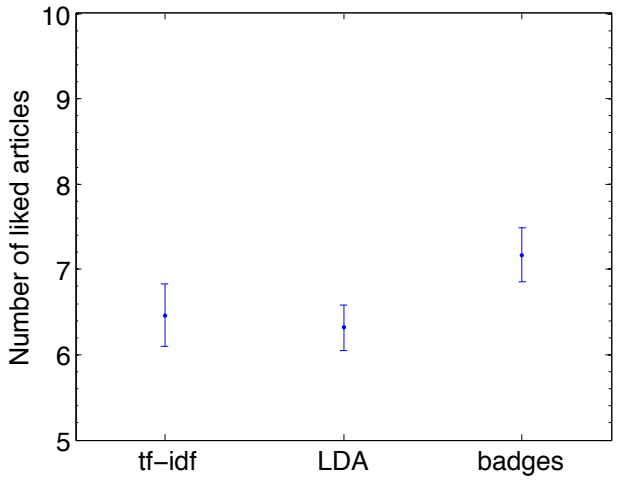


Figure 6.14: Results from the news recommendation user study, showing that the badge-based concept representation developed in this chapter leads to better article recommendations than competing document representations.

LDA on this fundamental news recommendation task. On average, users find the articles we recommend to them to be more interesting than the articles recommended via the competing document representations. This is what we expected, and backs our hypothesis that the badge-based representation is a preferable document representation for personalization tasks–particularly ones that cut across periods of time. While tf-idf is excellent at detecting article similarity within a time period, it is worse at detecting similar articles from two completely different periods of time. Meanwhile, LDA is at the mercy of a successful bipartite matching. The badge-based representation can overcome both challenges, leading to improved performance.

6.5 Related Work

Modern research on concept representations can be traced back to the psychology community, where there is a large body of work about how people represent concepts [Murphy, 2002]. For instance, early work in cognitive science considered featural representations of categories (e.g., people might remember that bears often have the features “fur” and “claws”) [Hampton, 1979, Hayes-Roth and Hayes-Roth, 1977]. Additionally, a large subfield of classical artificial intelligence is that of knowledge representation [Brachman and Levesque, 2004] and ontology learning [Gruber, 1993]. While related in principle, these approaches tend to be tailored to more formal reasoning, e.g., for logic-based reasoning or automation. In this chapter, we endeavored to develop a more flexible and fluid representation of concepts.

The idea of inferring information about the documents from their readers is not new; there is a rich line of research on *collaborative filtering*, which classifies, filters, or recommends documents by detecting readership patterns which, in some sense, represent the collaborative effort of all the readers [Sugiyama et al., 2004]. Main approaches for collaborative filtering, such as matrix completion [Candès and Tao, 2010], leverage the intuition that similar readers tend to read similar documents and predict a user’s preference for a new article by taking people who have already read the new article and comparing the current user’s past behavior against the past behavior of the other people. This approach unfortunately requires an article to

have a large number of readers before we can infer any meaningful information about it; in contrast, our approach gets around this problem by associating the user preferences with the *content* of the articles and thus can be used to analyze articles which have never been read.

Popular methods for collaborative filtering often assume low-dimensional latent factors in the readership patterns. For example, one can assume that a user's movie preferences can be explained by a small number of unreported reasons such as having favorite genres or favorite directors. Our approach also involves latent variables but guides the latent variable discovery by associating each latent factor with badge. As a result, our model can handle a large number of the latent factors without sacrificing much computational or statistical efficiency.

Because the latent factors in our model associate user preferences with topics in the document contents, our work draws upon the massive existing literature on topic modeling. From latent semantic indexing [Deerwester, 1988] to latent Dirichlet allocation [Blei et al., 2003], topic models try to capture the content of the documents through thematically focused and weighted collections of words known as *topics*. There are countless extensions to topic models and the labeled LDA model [Ramage et al., 2009] in particular presents a method of associating each of the latent topics with tags from the social bookmarking website `delicious.com`. Though it is reasonable to try to use the Labeled LDA algorithm to involve badges in a latent topic model, we use our algorithm to better promote sparsity and incorporate badge relations.

6.6 Conclusions

In this chapter, we took the challenges that we observed with document representations commonly used in personalization, and addressed them by creating a new concept representation that is effective at representing user interests over time. By building upon the ideas of transparency introduced in the previous chapter, we produced a document representation that can naturally lead to interpretable answers to the common user question: why did I get this recommendation? Moreover, the badge-based concept representation developed in this chapter facilitates insightful analysis of written documents by analyzing the attributes of likely readers.

However, some challenges remain:

- Not every word that a user writes in his or her Twitter profile should be considered worthy of being a badge. Deeper linguistic analysis of user profiles will be necessary to identify words or phrases that are most suited to representing user attributes.

- The simple single word badges we gathered from Twitter users works well for news recommendation, but how do we transfer this success to other information retrieval settings? Determining the equivalent of Twitter profiles for other domains, such as scientific research, is an open question of research.

Despite these challenges, our extensive empirical evaluation shows that the novel concept representation introduced in this chapter is ultimately useful for personalization, advancing the state-of-the-art in representing documents.

6.7 Appendix: Data Processing

In this appendix, we detail the steps of our entire data processing pipeline, for reproducibility of our experiments.

Initial Processing

1. Download three months worth of tweets from the Twitter Garden Hose: September 2010, September 2011 and September 2012.
2. Filter the tweets and extract those that are: (1) a tweet of a link; and, (2) came from a user with a non-empty profile. This leaves us with over 120 million tweets across the three months (cf. Table 6.1).
3. Filter the tweeted links to match one of 20,000 mainstream news sources, as defined by Google News. Additionally, we make sure the url ends in one of the following top level domains, to filter away content unlikely to be in English: {.com, .uk, .au, .ca, .us}.
4. Remove articles from *The Guardian*, as they will be in our test set.
5. Download each tweeted news article, and remove the html boilerplate using Boilerpipe.[13]
6. Process each article by:
    - Removing all non-alphanumeric characters;
    - Filtering out stopwords;
    - Filtering out words shorter than 3 characters or longer than 25 characters;
    - Filtering out words that contain any non-English characters (fast, but imperfect, way to filter out most foreign language words).
    - Stem words using the Porter stemming algorithm.

Vocabulary Selection At this point we proceed to select a vocabulary of words for each time period:

1. Ignore documents that have fewer than ten unique words, as well as documents where all the words appear only once.
2. Define the vocabulary to be all words that appear in at least 0.01% of all remaining tweeted news articles.

We reiterate that we have a separate vocabulary per time period. Also, as visualizing words from a stemmed vocabulary is not aesthetically pleasing, we take the convention in this chapter of displaying the most common unstemmed word for each word stem.

[13] `http://code.google.com/p/boilerpipe/`

Badge Selection In our experiments, we use badges that are single words. However, it is possible (and more desirable) to expand our definition of badges to include noun phrases, named entities or other syntactic constructs. Here, we compute statistics that will help us select a set of badges for each time period:

1. Extract each word appearing in any Twitter user profile from our training data set. We call these *badges*.
2. Stem the badges using the Porter stemming algorithm.
3. For each badge b, compute its cumulative weight across the training data by summing over all user profiles u, giving credit inversely proportional to the number of badges in a profile:

$$\sum_u \frac{\lambda_b^{(u)}}{\sum_a \lambda_a^{(u)}}.$$

4. Keep badges whose cumulative weight is at least 0.002% of the total number of tweeted articles in that time period.

Again, we have a separate badge set for each time period.

6.8 Appendix: Optimization

6.8.1 Dictionary Learning

In our projected stochastic gradient descent algorithm for learning our badge dictionary $\mathbf{B}$, we use the following parameters:

- Minibatch size $m = 500$.
- Our termination condition is based on a tolerance of $\epsilon = 10^{-8}$, or a maximum number of iterations of 10,000.
- Our initial step size parameter $\alpha_0 = 20$.
- Our sparsity regularization parameter $\lambda_B = 10^{-5}$.

We initialize $\mathbf{B} = \mathbf{Y}\mathbf{\Theta}^\mathsf{T}$, and then sparsify it, keeping the largest 200 values in each column, before renormalizing each column to have unit ℓ_2-norm.

6.8.2 Coding the Documents

We run the smoothed proximal gradient algorithm of Chen et al. [2012] to optimize Eq. (6.8), allowing us to code our test set of articles from *The Guardian*. We use three different settings of the regularization parameters:

1. **Heavily fused:** $\lambda_\mathcal{G} = 0.001$ and $\lambda_\theta = 2 \times 10^{-4}$;
2. **Lightly fused:** $\lambda_\mathcal{G} = 2 \times 10^{-5}$ and $\lambda_\theta = 2 \times 10^{-4}$;
3. **No fusion:** $\lambda_\mathcal{G} = 0$ and $\lambda_\theta = 2 \times 10^{-4}$.

We take our final concept representation to simply be the average of the codings generated by these three models. We found this set up to be most successful when running experiments on a separate validation data set.

6.9 Appendix: Experimental Details

Here are the fourteen political columnists we analyzed during our case study, and their July 2012 articles that we downloaded for analysis:

David Brooks

- http://www.nytimes.com/2012/07/13/opinion/brooks-why-our-elites-stink.html
- http://www.nytimes.com/2012/07/17/opinion/brooks-more-capitalism-please.html
- http://www.nytimes.com/2012/07/20/opinion/brooks-where-obama-shines.html
- http://www.nytimes.com/2012/07/24/opinion/brooks-more-treatment-programs.html
- http://www.nytimes.com/2012/07/27/opinion/brooks-the-olympic-contradiction.html
- http://www.nytimes.com/2012/07/31/opinion/brooks-dullest-campaign-ever.html

Ann Coulter

- http://www.anncoulter.com/columns/2012-07-04.html
- http://www.anncoulter.com/columns/2012-07-11.html
- http://www.anncoulter.com/columns/2012-07-18.html
- http://www.anncoulter.com/columns/2012-07-25.html

Maureen Dowd

- http://www.nytimes.com/2012/08/01/opinion/dowd-gadding-of-a-gawky-gowk.html
- http://www.nytimes.com/2012/07/29/opinion/sunday/dowd-mitts-olympic-meddle.html
- http://www.nytimes.com/2012/07/25/opinion/dowd-hiding-in-plain-sight.html
- http://www.nytimes.com/2012/07/22/opinion/sunday/dowd-paterno-sacked-off-his-pedestal.html
- http://www.nytimes.com/2012/07/18/opinion/dowd-whos-on-americas-side.html
- http://www.nytimes.com/2012/07/15/opinion/sunday/dowd-the-boy-who-wanted-to-fly.html
- http://www.nytimes.com/2012/07/08/opinion/sunday/cowboys-and-colleens.html
- http://www.nytimes.com/2012/07/04/opinion/gaelic-guerrilla.html
- http://www.nytimes.com/2012/07/01/opinion/sunday/the-wearing-of-the-green.html

Tom Friedman

- http://www.nytimes.com/2012/08/01/opinion/friedman-why-not-in-vegas.html
- http://www.nytimes.com/2012/07/29/opinion/sunday/friedman-coming-soon-the-big-trade-off.html
- http://www.nytimes.com/2012/07/25/opinion/friedman-syria-is-iraq.html
- http://www.nytimes.com/2012/07/22/opinion/sunday/friedman-the-launching-pad.html
- http://www.nytimes.com/2012/07/04/opinion/what-does-morsi-mean-for-israel.html

- http://www.nytimes.com/2012/07/01/opinion/sunday/taking-one-for-the-country.html

Jonah Goldberg

- http://www.nationalreview.com/articles/304711/live-free-and-uninsured-jonah-goldberg
- http://www.nationalreview.com/articles/304819/politics-and-symptoms-sick-culture-jonah-goldberg
- http://www.nationalreview.com/articles/308431/blame-barclays-not-capitalism-jonah-goldberg
- http://www.nationalreview.com/articles/309299/tilting-un-windmill-jonah-goldberg
- http://www.nationalreview.com/articles/309736/romney-and-bain-outsourcing-hysteria-jonah-goldberg
- http://www.nationalreview.com/articles/310080/co-sponsoring-your-success-jonah-goldberg
- http://www.nationalreview.com/articles/311235/brian-ross-s-brain-cramp-jonah-goldberg
- http://www.nationalreview.com/articles/312417/colorado-and-case-capital-punishment-jonah-goldberg

David Ignatius

- http://www.washingtonpost.com/opinions/david-ignatius-irans-bargaining-position-hardens/2012/07/02/gJQAi5NOJW_story.html
- http://www.washingtonpost.com/opinions/david-ignatius-israels-arab-spring-problem/2012/07/05/gJQAV5JrRW_story.html
- http://www.washingtonpost.com/opinions/david-ignatius-can-diplomacy-succeed-with-iran-and-syria/2012/07/11/gJQA7LwzdW_story.html
- http://www.washingtonpost.com/opinions/david-ignatius-pakistan-us-have-a-neurotic-relationship/2012/07/13/gJQABEDoiW_story.html
- http://www.washingtonpost.com/opinions/david-ignatius-syria-approaches-the-tipping-point/2012/07/18/gJQAFoCvtW_story.html
- http://www.washingtonpost.com/opinions/david-ignatius-central-banks-face-a-giant-bill-coming-due/2012/07/20/gJQALdJsyW_story.html
- http://www.washingtonpost.com/opinions/david-ignatius-the-day-after-in-syria/2012/07/25/gJQA4Uey9W_story.html
- http://www.washingtonpost.com/opinions/david-ignatius-senates-anti-leaking-bill-doesnt-address-the-real-sources-of-information/2012/07/31/gJQAPBElNX_story.html

Joe Klein

- http://swampland.time.com/2012/07/23/gunclingers-aurora-assault-weapons-and-the-rise-of-mass-shootings/
- http://swampland.time.com/2012/07/19/latest-column-93/
- http://swampland.time.com/2012/07/18/who-built-i-80/
- http://swampland.time.com/2012/07/15/inconvenient-truths/
- http://swampland.time.com/2012/07/13/bained/

- http://swampland.time.com/2012/07/11/a-friend-remembered/
- http://swampland.time.com/2012/07/02/you-say-tomato-i-call-bullpucky/

Charles Krauthammer

- http://www.washingtonpost.com/opinions/charles-krauthammer-the-imperial-presidency-revisited/2012/07/05/gJQAR66PQW_story.html
- http://www.washingtonpost.com/opinions/charles-krauthammer-the-islamist-ascendancy/2012/07/12/gJQArj9PgW_story.html
- http://www.washingtonpost.com/opinions/charles-krauthammer-did-the-state-make-you-great/2012/07/19/gJQAbZOiwW_story.html
- http://www.washingtonpost.com/opinions/charles-krauthammer-why-hes-going-where-hes-going/2012/07/26/gJQAGkzJCX_story.html
- http://www.washingtonpost.com/opinions/charles-krauthammer-busted-mr-pfeiffer-and-the-white-house-blog/2012/07/29/gJQA8M46IX_story.html

Nicholas Kristof

- http://www.nytimes.com/2012/07/01/opinion/sunday/africa-on-the-rise.html
- http://www.nytimes.com/2012/07/05/opinion/doughnuts-defeating-poverty.html
- http://www.nytimes.com/2012/07/08/opinion/sunday/the-coffin-maker-benchmark.html
- http://www.nytimes.com/2012/07/12/opinion/kristof-obamas-fantastic-boring-idea.html
- http://www.nytimes.com/2012/07/26/opinion/kristof-safe-from-fire-but-not-gone.html
- http://www.nytimes.com/2012/07/29/opinion/sunday/kristof-blissfully-lost-in-the-woods.html

William Kristol

- http://www.weeklystandard.com/articles/only-108-days-go_648828.html
- http://www.weeklystandard.com/articles/campaign-altogether-old_648556.html
- http://www.weeklystandard.com/articles/profiles-courage_648224.html
- http://www.weeklystandard.com/articles/obama-retreat_647776.html

Paul Krugman

- http://www.nytimes.com/2012/07/06/opinion/off-and-out-with-mitt-romney.html
- http://www.nytimes.com/2012/07/09/opinion/krugman-mitts-gray-areas.html
- http://www.nytimes.com/2012/07/13/opinion/krugman-whos-very-important.html
- http://www.nytimes.com/2012/07/16/opinion/krugman-policy-and-the-personal.html
- http://www.nytimes.com/2012/07/20/opinion/krugman-pathos-of-the-plutocrat.html
- http://www.nytimes.com/2012/07/23/opinion/krugman-loading-the-climate-dice.html
- http://www.nytimes.com/2012/07/27/opinion/money-for-nothing.html
- http://www.nytimes.com/2012/07/30/opinion/krugman-crash-of-the-bumblebee.html

Kathleen Parker

- http://www.washingtonpost.com/opinions/kathleen-parker-the-ladies-of-mount-vernon-have-preserved-washingtons-home/2012/07/03/gJQART6dLW_story.html
- http://www.washingtonpost.com/opinions/kathleen-parker-south-carolina-politics-gets-insulting/2012/07/06/gJQArcAfSW_story.html
- http://www.washingtonpost.com/opinions/kathleen-parker-doug-marlette-a-friend-remembered/2012/07/10/gJQAj8JfbW_story.html
- http://www.washingtonpost.com/opinions/kathleen-parker-romneys-critics-say-the-silliest-things/2012/07/13/gJQAXMNoiW_story.html
- http://www.washingtonpost.com/opinions/kathleen-parker-how-to-get-smart-news-literacy-programs-train-readers-to-look-beyond-infotain
2012/07/17/gJQAY1m2rW_story.html
- http://www.washingtonpost.com/opinions/kathleen-parker-obama-campaign-shows-its-desperation-in-romney-attack/2012/07/20/gJQAiYCsyW_story.html
- http://www.washingtonpost.com/opinions/kathleen-parker-in-poland-romney-addresses-economic-and-religious-freedom/2012/07/31/gJQA2c7kNX_story.html

Frank Rich

- http://nymag.com/daily/intel/2012/07/frank-rich-mitt-cant-wait-out-his-tax-storm.html
- http://nymag.com/daily/intel/2012/07/frank-rich-romney-has-a-tax-and-koch-problem.html
- http://nymag.com/news/frank-rich/declining-america-2012-7/

Fareed Zakaria

- http://fareedzakaria.com/2012/07/26/failure-to-launch/
- http://fareedzakaria.com/2012/07/18/what-voters-are-really-choosing-in-november/
- http://fareedzakaria.com/2012/07/12/tax-and-spend/
- http://fareedzakaria.com/2012/07/05/curbing-the-cost-of-health-care/

Chapter 7

Conclusion

7.1 Thesis Summary

As described at the start of this thesis, we are today firmly in an era of information overload and "Big Data." From online news to scholarly research to multimedia, we are being inundated with information on a daily basis, and, by and large, we are struggling to extract from it actionable knowledge. One potential approach for addressing this challenge lies in *personalizing* results to the tastes of individual users. Different people have different preferences when it comes to books, news stories, movies and web pages, and in all of these domains, personalization has played an integral role in providing relevant recommendations. However, despite some success, personalization faces significant challenges of its own:

1. How do we recommend items to new users? How do we recommend new items to existing users? This is the classic *cold start problem*.

2. How do we take into account multiple—perhaps orthogonal—dimensions of user preferences? For example, a researcher might be interested in a particular topic area (e.g., kernel methods), and prefers particular publications (e.g., the *Journal of Machine Learning Research*), and likes articles that have many proofs. How do we combine these dimensions in a principled manner?

3. Can we provide mechanisms for *rich user interaction*, such that a user can express her information need in the most natural of ways?

4. How do we model the dependencies between recommended items, such that we *avoid redundancy*? A reader may be interested in Egypt, but is unlikely to enjoy five articles in a row about the devaluation of the Egyptian pound.

5. How do we make personalization more *transparent*, such that users have a clearer picture of how they are perceived, with the ability to make corrections as necessary?

In this thesis, we set out to address all of these problems by designing a general framework for personalization, which we call *interactive concept coverage*. This framework involves framing personalized recommendation as a probabilistic budgeted max-cover problem, where each item to be recommended is defined to *probabilistically cover* one or more *concepts*. From user interaction, we define weights on concepts and affinities for items, with the hope of achieving personalized, diverse recommendations.

The goal of our thesis research was to validate the hypothesis that this interactive concept coverage methodology can successfully address the problems with personalization described above, leading us to the following thesis statement:

> In a variety of information overload settings, the Interactive Concept Coverage framework produces personal recommendations that are highly relevant, diverse and transparent, incorporating complex user preferences through rich user interaction.

In Chapter 3, we motivated and defined the notion of *probabilistic budgeted max-cover*, as applied to the recommendation of blog posts and news articles. In this queryless setting, we described an online learning algorithm for learning personalized user preferences over content from simple user feedback. A user study and offline experiments showed the success of our approach at recommending relevant and diverse news articles, outperforming competition from Google, Digg and others.

In Chapter 4, we extended our approach to a setting with complex user interactions. Using the domain of scientific literature discovery as a case study, we described how a rich user query—beyond simple keywords—can be seamlessly incorporated into our framework. Here, we define a notion of coverage that is not exclusively text-based, allowing us to take advantage of the richer structure of the problem. Moreover, we define for the first time an affinity function over the items to be recommended, encoding the trust preferences individual researchers have over different authors. Finally, an extensive user study on computer science researchers showed that our methodology, when applied in this setting, significantly outperformed state-of-the-art algorithms and Google Scholar, recommending more useful, diverse and trustworthy articles.

In Chapter 5, we shifted to the important (but understudied) issue of transparency in personalization. We describe an approach for predicting interpretable user attributes from activity on the microblogging site Twitter. Specifically, we leverage the words that users use to publicly describe themselves—so-called *badges*—to build a transparent user model suitable for personalization. Chapter 6 builds upon this work by using badges to construct a concept representation that allows us to describe documents by their likely readers. We show that this representation has many desirable properties beyond human interpretability, including being able to deal with concepts that change over time. Returning to the news recommendation setting of Chapter 3, we show that, when powered by this badge-based concept representation, our interactive concept coverage methodology outperforms alternative concept representations on a variety of online and offline metrics.

Fundamentally, over the course of this thesis, we showed that, by using the interactive concept coverage methodology, we were able to successfully beat both state-of-the-art algorithms and market leaders on two important personalization domains: news recommendation and scientific research. In the process, we successfully addressed each of the six aforementioned challenges of personalization, thereby validating our hypothesis.

7.2 Recommendations

While working on this thesis, lessons were learned that proved to be useful again and again throughout the research process. In this section, we briefly describe three such lessons that stood out and may have lasting benefit beyond this work.

Diversity should not be an afterthought.
From web search to personalized news to general recommendation systems, it is tempting to leave diversity for the end, as something to be addressed after the fact. For example, a simple and common approach for recommendation (and web search) is to design a ranking function over the items to be recommended, where each item is scored independently, and then sort the items by their scores. If diversity is a concern, it is treated as a post-processing step, perhaps with some local clustering. Instead, in every occasion, we found that by directly modeling the intuitive notion that users get tired of redundant results, we achieve results that are not just more diverse, but also more relevant.

Hence, a good starting point for a designer of a personalization system is to think of examples of redundant result sets, and ask: what exactly is it about these items that makes them redundant? The answer to this question will help define an appropriate concept representation that can be directly used in our framework. For example, if the items are documents, and the redundancy is just related to the topical content, then perhaps words are reasonable concepts to be used (cf. Chapter 3). However, if redundancy stems not just from the content of the items, but from, e.g., how they relate to a particular query, then a more complex concept representation may be preferable (cf. Chapter 4).

Optimize for the prize.
The research in this thesis was driven by real-world problems, from news recommendation to scientific discovery. In our evaluations, as we compared our methodology to competing approaches, we came to a realization that is perhaps obvious in hindsight: algorithms that directly optimize for the task at hand tend to outperform algorithms that treat the task as a side effect. For example, in Chapter 4, the relational topic model (RTM) [Chang and Blei, 2010], as a generative, probabilistic model over a linked set of documents, can easily be adapted to our setting of recommending articles based on a query set of relevant documents. However, despite this ease of use, the primary purpose of an RTM (and other topic models) is tangential to our task; rather than providing meaningful paper recommendations, an RTM is designed to provide an accurate, statistical fit to a corpus of linked documents. We believe that this divergence of purpose is responsible for the poor performance of the RTM on our task.

Thus, when faced with a real-world problem, rather than taking an off-the-shelf model and trying to immediately apply it, a better first step is to write down an objective function that, if optimized, will directly lead to an appropriate solution to the problem. This is the approach we used with great success throughout this thesis.

Think creatively about evaluation, data and user interactions.
Out of thousands of papers on topic modeling, most likely use the same held-out log likelihood evaluation metric called perplexity. For years, machine learning researchers have reported results on the same standardized UCI data sets. As described in Chapter 1 of this thesis, keyword search is the predominant form of user interaction in information retrieval research and practice. If we had limited ourselves to standard evaluations, canned data sets and pre-existing user interactions, our research would have been handicapped from the start. Thus, perhaps the most critical lesson we learned throughout this thesis work is to think creatively about these three factors.

- *Evaluation:* This thesis is about personalization, and as a result, throughout this work, we conducted user studies to obtain meaningful evaluations of our techniques. Such studies are often difficult and their results unpredictable, but testing a hypothesis involving personal preferences is difficult to answer without obtaining real personal preferences from real users. An important lesson of our work is that much time should be given to thinking carefully and creatively about how to test one's ideas.

- *Data:* Some of the most exciting results in this thesis are due to a creative look at a common data set (cf. Chapters 5 and 6). Many papers have been written on Twitter, but nearly all have ignored the short user profile at the top of each user's Twitter page. A key lesson here is that a good algorithm may not be good enough without the right data. Hence, much of the effort in this thesis work was devoted to finding, obtaining and processing the right data.
- *User interaction:* The title of this thesis starts with the phrase, "beyond keyword search." In our work (and particularly in Chapter 4), we found that, by freeing ourselves from solely thinking in terms of keyword search, we were able to devise rich user interactions that allowed users to express their complex information needs more naturally and efficiently.

We believe that the work conducted over the course of this thesis research—both the general methodology and the specific case studies—makes a useful contribution to the state of the art in personalization. Chapter 8 closes this thesis by describing open questions and remaining challenges.

Chapter 8

Future Work

In this chapter, we describe ideas for future research, inspired by our work in this thesis.

8.1 Concept Hierarchies and Cuts Over Time

The interactive concept coverage methodology that we introduce in this thesis revolves around defining a concept representation for the items to be recommended. In our framework, the main source of personalization comes from learning weights over a set of concepts, based on user interaction. We assume that the set of concepts is fixed, common to all users, but the concept weights are what define individual preferences. Alternatively, we can imagine a scenario where the concepts themselves differ from person to person; the set of concepts associated with a user is what drives personalization.

For example, we might consider the case of an avid sports fan. To him, the Pittsburgh Steelers, LeBron James and Wimbledon all represent distinct concepts coming from the world of sports. However, another user who may not care so much about sports may consider these all to be equivalent from his point of view. Namely, for this user, an article about the Steelers might just as well be an article about tennis; it's all "just sports" to him. From this example, we might conclude that each user interacts with concepts at different levels of granularity, and that a faithful model of user interests would take such a phenomenon into account.

One potential avenue is to assume that all concepts in the world lie along a hierarchy, and that a user's preferences are defined by: (1) a cut in the hierarchy; and, (2) a weight vector over the concepts along the cut. Continuing our example, we can imagine a hierarchy that groups together many sports-related concepts. Thus, the concept cut for a sports fan will perhaps include the leaves of this part of the tree. However, another user who is not interested in sports will cut this part of the hierarchy at a much higher level. The limited interests and attention of a user can be modeled by enforcing a constraint on the size of the cut.

Many open questions exist in this setting:

- If we are given a tree of concepts, can we efficiently learn a user's cut in the hierarchy?
- How do we learn a hierarchy over concepts, and should it be limited to a tree?

- Does such a user model improve performance in any standard recommendation tasks?

A particularly important open question in personalization is how to model the temporal dynamics of user interests. A user might be interested in tennis only during a Grand Slam, but that interest wanes quickly thereafter. Likewise, a consumer buying a digital camera loses interest in camera recommendations immediately upon purchase. One approach for answering this question lies in allowing the cut in the concept hierarchy to change over time, such that when a user is deeply interested in an area, her cut occurs at a low level in the hierarchy, but as interest wanes, the cut recedes to a higher level.

8.2 Modeling the Knowledge Remainder

Continuing our theme of bringing transparency to personalization, one intriguing line of research involves informing users about how a recommended set of items relates to the entire set of items from which they were picked. For example, if a researcher is recommended a set of scientific papers about a particular area—say, Bayesian nonparametrics—how does she know what else is out there? It could be that most of the remaining Bayesian nonparametrics papers in the corpus are similar to the ones already presented, or, alternatively, the recommended papers could represent just a narrow sliver of the total body of work. Such information could determine whether the researcher is ultimately satisfied by the results, or feels that something must be missing. Similar situations can occur in many other domains, from online shopping ("did I really see all the different types of cameras?") to planning a vacation ("do I have a good idea of all there is to do in Barcelona?").

These scenarios are not just hypothetical; recent work in distributed sensemaking has experimentally verified such behavior, showing that, when presented with knowledge maps created by others, e.g., on how to plant a garden, users immediately seek out any materials discarded by the map creators, to ensure that they are not missing anything important [Fisher et al., 2012].

Thus, a useful research contribution straddling machine learning and human computer interaction would devise a method for providing users with a succinct, interpretable description of what else is left—a so-called *knowledge remainder*.

8.3 Automatic Fact Checking of the Web

In Chapter 4 of this thesis, we model trust patterns among scientific researchers by analyzing the citation graph. An interesting question for future research is how to extend such a trust model to other settings—in particular, online news. Without the luxury of a citation graph, we might consider analyzing tweets and retweets of different news sites on Twitter. Following Chapter 6, we can associate trust with badges, leading to statements such as, "By a 9 to 1 margin, liberals share links from `motherjones.com` over conservatives."

An ultimate goal for such a research direction would be to provide an automatic fact checking of the Web. A reader seeing a suspicious sentence claiming that Barack Obama is a Muslim born in Kenya should be able to highlight it, and gain a report on its expected veracity.

Beyond the details of how a statement's veracity might be represented, this line of work leads to more fundamental questions about what truth really means. In particular, how differently do various people view

the truth of a statement? How do people decide that something is true or not? While these philosophical questions are certainly abstract, contemplating them can shed light on the types of user interactions that would be most meaningful.

8.4 Interactive Concept Coverage Beyond Text

In this thesis, we validated our interactive concept coverage methodology on two distinct domains: news recommendation and scientific literature discovery. While different in many ways, these two domains are both text-based. An interesting—and immediate—line of future work would explore how this methodology can be extended to personalizing non-text content, such as images, videos or friends on a social network.

By following the general recipe of our framework, the key questions include:

- What is it that makes images (or videos or friends) redundant in a particular retrieval setting?
- Are there novel user interactions that are unique to these settings that can be useful for preference learning?
- Is it easier or harder to evaluate our performance in these settings?

8.5 Richer User Interactions

Throughout this thesis, we considered a variety of user interactions, from simple likes and dislikes (Chapter 3) to richer forms of queries and feedback (Chapter 4). However, these are just first steps down a path towards more complex user interactions. For example, rather than model user sessions in a one-shot fashion, it is more realistic to assume that users will iterate with an information retrieval system, providing many queries in one sitting in an attempt to find the particular nugget of information that they are seeking. An important area of future work is to build upon early work in this area (e.g., [Pandit and Olston, 2007]) and consider how such rich interactions can be utilized in our framework and beyond.

In the same vein, it is important to note that the approaches described in this thesis all deal with *absolute* feedback: how highly does a user rate a particular item? Previous work has shown that such absolute feedback can be unreliable, and that *relative* feedback is often preferable in information retrieval settings [Joachims et al., 2007]. In future work, it would be instructive to consider how relative feedback can be incorporated into our framework, perhaps based on novel user interactions. For example, rather than present a set of items to a user that is selected based on a single setting of concept weights, we might interleave two sets of documents, each selected based on a different concept weighting, allowing us to learn from relative feedback. Moreover, we might also consider clustering-based feedback, that allows users to directly describe which items might be redundant to each other.

Finally, an additional direction of research in this area is to take a broader look at user interaction, in an attempt to directly model a user's *utility*. For example, can we measure the utility of providing diversity? What does it mean from a utility perspective when six out of ten presented results are "liked" as opposed to five out of ten? Answering questions like these depends on the particular retrieval setting and application, and is a promising area of future work.

Bibliography

Robert Adler, John Ewing, and Peter Taylor. Citation statistics. *Statistical Science*, 24(1):1–14, 2009. 4.4

Raja H. Affandi, Alex Kulesza, and Emily B. Fox. Markov determinantal point processes. In *28th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2012. 3.6.1

Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, Raghu Ramakrishnan, Nitin Motgi, Scott Roy, and Joe Zachariah. Online models for content optimization. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *Advances in Neural Information Processing Systems (NIPS) 21*, 2009. 3.6.2

Amr Ahmed, Mohamed Aly, Joseph Gonzalez, Shravan Narayanamurthy, and Alexander J. Smola. Scalable inference in latent variable models. In *Fifth ACM International Conference on Web Search and Data Mining (WSDM)*, 2012a. 5.4

Amr Ahmed, Choon Hui Teo, S.V.N. Vishwanathan, and Alexander J. Smola. Fair and balanced: Learning to present news stories. In *Fifth ACM International Conference on Web Search and Data Mining (WSDM)*, 2012b. 3.5

Edoardo M. Airoldi, Elena A. Erosheva, Stephen E. Fienberg, Cyrille Joutard, Tanzy Love, and Suyash Shringarpure. Reconceptualizing the classification of PNAS articles. *Proceedings of the National Academy of Sciences*, 107(49):20899–20904, 2010. 4.6

David Andrzejewski, Xiaojin Zhu, Mark Craven, and Ben Recht. A framework for incorporating general domain knowledge into latent Dirichlet allocation using first-order logic. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2011. 5.3

Artin Armagan, David B. Dunson, and Merlise Clyde. Generalized beta mixtures of Gaussians. In *Advances in Neural Information Processing Systems (NIPS)*, 2011. 2.2.2

Albert-László Barabási. On the topology of the scientific collaboration networks. *Physica A*, 311:590–614, 2002. 4.6

Marcia J. Bates. The design of browsing and berrypicking techniques for the online search interface. *Online Review*, 13:407–424, 1989. 4.6

Martin Bland. *An Introduction to Medical Statistics*. Oxford Medical Publications, 3rd edition, 2000. 11

David M. Blei and John Lafferty. Dynamic topic models. In *23rd International Conference on Machine Learning (ICML)*, 2006. 4.6, 6.1

David M. Blei and John Lafferty. A correlated topic model of *Science*. *Annals of Applied Statistics*, 1:17–35, 2007. 4.6

David M. Blei and John Lafferty. Topic models. In A. Srivastava and M. Sahami, editors, *Text Mining: Theory and Applications*. Taylor and Francis, 2009. 1.2, 2.2, 3.1, 4.6, 5.1, 6

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3: 993–1022, 2003. 3.1, 1, 5.1, 6.1, 6.5

Kurt D. Bollacker, Steve Lawrence, and C. Lee Giles. Discovering relevant scientific literature on the Web. *IEEE Intelligent Systems and their Applications*, 15:42–47, 2000. 4.6

Ronald J. Brachman and Hector J. Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann, 2004. 6.5

Joseph K. Bradley, Aapo Kyrola, Danny Bickson, and Carlos Guestrin. Parallel coordinate descent for L1-regularized loss minimization. In *28th International Conference on Machine Learning (ICML)*, 2011. 6.3.2

Emmanuel J. Candès and Terence Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010. 6.5

Emmanuel J. Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006. 2.2.1

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based re-ranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 1998. 3.6.1

Carlos M. Carvalho, Nicholas G. Polson, and James G. Scott. Handling sparsity via the Horseshoe. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009. 2.2.2

George Casella and Edward I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, August 1992. 2.1.2

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning and games*. Cambridge University Press, 2006. 3.3.3

Jonathan Chang. Collapsed Gibbs sampling for topic models. `http://cran.r-project.org/web/packages/lda/`, 2010. 4.5

Jonathan Chang and David M. Blei. Hierarchical relational models for document networks. *Annals of Applied Statistics*, 4: 124–150, 2010. 4.5, 7.2

Harr Chen and David Karger. Less is more. In *29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2006. 3.6.1

Patrick Chen, Huafeng Xie, Sergei Maslov, and Sidney Redner. Finding scientific gems with Google. *Journal of Informetrics*, 1: 8–15, 2007. 4.4

Shaobing Chen and David L. Donoho. On basis pursuit. Technical report, Stanford University, 1994. 2.2.1

Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1): 129–159, 2001. 2.2.1

Xi Chen, Qihang Lin, Seyoung Kim, Jaime Carbonell, and Eric P. Xing. Smoothing proximal gradient method for general structure sparse regression. *Annals of Applied Statistics*, 6(2):719–752, 2012. 2.2.1, 6.3.3, 6.8.2

Gregory F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990. 2.1.2

Gianmarco De Francisci Morales, Aristides Gionis, and Claudio Lucchese. From chatter to headlines: harnessing the real-time web for personalized news recommendation. In *Fifth ACM International Conference on Web Search and Data Mining (WSDM)*, 2012. 6.1

Derek J. de Solla Price. Networks of scientific papers. *Science*, 149:510:515, 1965. 4.6

Scott Deerwester. Improving information retrieval with latent semantic indexing. In *Proceedings of the 51st ASIS Annual Meeting*, 1988. 6.5

Denis Diderot. *Encyclopedia, or a systematic dictionary of the sciences, arts and crafts*. Briasson, David, Le Breton, and Durand, Paris, 1755. 1

Laura Dietz, Steffen Bickel, and Tobias Scheffer. Unsupervised prediction of citation influences. In *24th International Conference on Machine Learning (ICML)*, 2007. 4.6

Khalid El-Arini, Gaurav Veda, Dafna Shahaf, and Carlos Guestrin. Turning down the noise in the blogosphere. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2009. 4.6, 6.1, 6.1

Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2008. 4.6

Elena A. Erosheva, Stephen E. Fienberg, and John Lafferty. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences*, 101:5220–5227, 2004. 4.6

Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the Internet topology. In *ACM SIGCOMM*, 1999. 4.4

Tim Finin, Anupam Joshi, Pranam Kolari, Akshay Java, Anubhav Kale, and Amit Karandikar. The information ecology of social media and online communities. *AI Magazine*, 2008. 3.6.2

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005. 3.4

Kristie Fisher, Scott Counts, and Aniket Kittur. Distributed sensemaking: Improved sensemaking by leveraging the efforts of previous users. In *ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2012. 8.2

Emily B. Fox. *Bayesian Nonparametric Learning of Complex Dynamical Phenomena*. PhD thesis, Massachusetts Institute of

Technology, 2009. 2.1.2, 2

Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29: 79–103, 1999. 3.3.3, 3.3.3, 3.8, 3.8

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A note on the group lasso and a sparse group lasso. *arXiv:1001.0736*, 2010. 2.2.1, 2.2.1

Arnoldo Frigessi, Patrizia Di Stefano, Chii-Ruey Hwang, and Shuenn-Jyi Sheu. Convergence rates of the Gibbs sampler, the Metropolis algorithm and other single-site updating dynamics. *Journal of the Royal Statistical Society, Series B*, 55(1):205–219, 1993. 2

Eugene Garfield. Citation analysis as a tool in journal evaluation. *Science*, 178:471–479, 1972. 4.4

Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984. 2.1.2, 2.1.2

Sean Gerrish and David M. Blei. A language-based approach to measuring scholarly impact. In *27th International Conference on Machine Learning (ICML)*, 2010. 4.6

Sean Gerrish and David M. Blei. Predicting legislative roll calls from text. In *28th International Conference on Machine Learning (ICML)*, 2011. 6.4.3

Zoubin Ghahramani and Katherine A. Heller. Bayesian sets. In B. Sch olkopf Y. Weiss and J. Platt, editors, *Advances in Neural Information Processing Systems (NIPS) 18*, 2006. 4.6

Walter R. Gilks, Sylvia Richardson, and David J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, 1995. 2.1.2

Jennifer Gillenwater, Alex Kulesza, and Ben Taskar. Discovering diverse and salient threads in document collections. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2012a. 3.6.1

Jennifer Gillenwater, Alex Kulesza, and Ben Taskar. Near-optimal MAP inference for determinantal point 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 (NIPS) 25*, 2012b. 3.6.1

Joseph Gonzalez, Yucheng Low, Haijie Gu, Danny Bickson, and Carlos Guestrin. PowerGraph: Distributed graph-parallel computation on natural graphs. In *Proceedings of Operating System Design and Implementation (OSDI)*, 2012. 5.4, 6.4.4

Nicole Greenstein. Negative ads: A shift in tone for the 2012 campaign. *TIME Magazine*, July 17, 2012. URL `http://swampland.time.com/2012/07/17/negative-ads-a-shift-in-tone-for-the-2012-campaign/`. 6.4.2

Jim E. Griffin and Philip J. Brown. Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis*, 5 (1):171–188, 2010. 2.2.2

Thomas L. Griffiths and Zoubin Ghahramani. Infinite latent feature models and the Indian buffet process. In Y. Weiss, B. Sch olkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 18, 2005. 5

Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101: 5228–5235, 2004. 1, 4.6

Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993. 6.5

James A. Hampton. Polymorphous concepts in semantic memory. *Journal of Verbal Learning and Verbal Behavior*, 18:441–461, 1979. 6.5

Chris Hans. Bayesian lasso regression. *Biometrika*, 96(4):835–845, 2009. 2.2.2

W. Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. 2.1.2

Barbara Hayes-Roth and Frederick Hayes-Roth. Concept learning and the recognition and classification of exemplars. *Journal of Verbal Learning and Verbal Behavior*, 16:321–328, 1977. 6.5

Jorge E. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences*, 102(46):16569–16572, 2005. 4.4

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963. 4.2.2

Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 1999. 6.3.1

Hemant Ishwaran and Lancelot F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, March 2001. 2.1.2

Hemant Ishwaran and J. Sunil Rao. Spike and slab variable selection: Frequentist and Bayesian strategies. *Annals of Statistics*, 33 (2):730–773, 2005. 2.2.2

Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group Lasso with overlap and graph Lasso. In *26th International Conference on Machine Learning (ICML)*, 2009. 2.2.1

Finn V. Jensen. *Bayesian Networks and Decision Graphs*. Springer, 2001. 2.1

Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in Web search. *ACM Transactions on Information Systems*, 25(2), April 2007. 8.5

Michael I. Jordan. Graphical models. *Statistical Science (Special Issue on Bayesian Statistics)*, 19:140–155, 2004. 2.1

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999. 2.1.2

Samir Khuller, Anna Moss, and Joseph Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1): 39–45, April 1999. 3.1, 3.2

Seyoung Kim, Kyung-Ah Sohn, and Eric P. Xing. A multivariate regression approach to association analysis of a quantitative trait network. In *Proceedings of the 17th Conference on Intelligent Systems for Molecular Biology (ISMB)*, 2009. 2.2.1, 6.3.3

Michael Kinsley. How many blogs does the world need? *TIME Magazine*, 172(22), December 2008. 3

Jon Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46:604–632, 1999. 4.4

Kwangmoo Koh, Seung-Jean Kim, and Stephen Boyd. An interior-point method for large-scale ℓ_1-regularized logistic regression. *Journal of Machine Learning Research*, 8:1519–1555, 2007. 2.2.1

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009. 2.1, 2.1.1, 2.1.2

Yehuda Koren, Robert M. Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer Magazine*, 42(8):30–37, August 2009. 5.3

Andreas Krause. *Optimizing Sensing: Theory and Applications*. PhD thesis, Carnegie Mellon University, 2008. 7

Andreas Krause and Carlos Guestrin. Optimizing sensing from water to web. *IEEE Computer Magazine*, August 2009. 3.6.1

Andreas Krause, H. Brendan McMahan, Carlos Guestrin, and Anupam Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 9:2761–2801, 2008. 3.6.1

Alex Kulesza. *Learning with determinantal point processes*. PhD thesis, University of Pennsylvania, 2012. 3.6.1

Alex Kulesza and Ben Taskar. Structured determinantal point processes. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems (NIPS) 23*, 2010. 3.6.1

Alex Kulesza and Ben Taskar. k-DPPs: fixed-size determinantal point processes. In *28th International Conference on Machine Learning (ICML)*, 2011a. 3.6.1

Alex Kulesza and Ben Taskar. Learning determinantal point processes. In *27th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2011b. 3.6.1

Ni Lao and William W. Cohen. Relational learning using a combination of path-constrained random walks. *Machine Learning*, 81 (1):53–67, 2010. 4.6

Steffen L. Lauritzen. *Graphical models*. Oxford University Press, 1996. 2.1

Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y. Ng. Efficient L_1 regularized logistic regression. In *21st AAAI Conference on Artificial Intelligence*, 2006. 2.2.1

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2007. 3.2, 3.6.1, 3.6.2, 4.3.2

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *19th International World Wide Web Conference (WWW)*, 2010. 6.1

Jun S. Liu. Peskun's theorem and a modified discrete-state Gibbs sampler. *Biometrika*, 83(3):681–682, 1996. 2

Yucheng Low, Joseph Gonzalez, Aapo Kyrola, Danny Bickson, Carlos Guestrin, and Joseph M. Hellerstein. Distributed GraphLab:

A framework for machine learning and data mining in the cloud. *Proceedings of Very Large Data Bases (PVLDB)*, 5(8): 716–727, 2012. 5.4, 6.4.4

Richard F. MacLehose and David B. Dunson. Bayesian semiparametric multiple shrinkage. *Biometrics*, 66(2):455–462, June 2010. 2.2.2

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010. 2.2.1

Sean M. McNee, Istvan Albert, Dan Cosley, Prateep Gopalkrishnan, Shyong K. Lam, Al Mamunur Rashid, Joseph A. Konstan, and John Riedl. On the recommending of citations for research papers. In *ACM Conference on Computer Supported Cooperative Work (CSCW)*, 2002. 6, 4.6

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physica*, 21(6):1087–1092, 1953. 2.1.2

David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2011. 6.4.4, 6.11

T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988. 2.2.2

Gregory L. Murphy. *The Big Book of Concepts*. MIT Press, 2002. 6.5

Balas K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, April 1995. 2.2.1

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978. 3.2

Mark E. J. Newman. Scientific collaboration networks: I. network construction and fundamental results. *Physical Review E*, 64: 016131, 2001a. 4.6

Mark E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98: 404–409, 2001b. 4.6

Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996. 2.2, 2.2.1

Christopher Olston and Ed H. Chi. ScentTrails: Integrating browsing and searching on the Web. *ACM Transactions on Computer-Human Interaction*, 10:177–197, 2003. 4.6

Larry Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: bringing order to the Web. Technical report, Stanford University InfoLab, 1999. 4.4

Shashank Pandit and Christopher Olston. Navigation-aided retrieval. In *16th International World Wide Web Conference (WWW)*, 2007. 4.6, 8.5

Eli Pariser. *The Filter Bubble: What the Internet Is Hiding from You*. Penguin Press, 2011. 5, 5

Trevor Park and George Casella. The Bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008. 2.2.2

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988. 2.1, 2.1.1, 2.1.2

Nicholas G. Polson and James G. Scott. Shrink globally, act locally: Sparse Bayesian regularization and prediction. *Bayesian Statistics*, 9, 2010. 2.2.2

Keith T. Poole and Howard Rosenthal. A spatial model for legislative roll call analysis. *American Journal of Political Science*, 29 (2):357–384, May 1985. 6.4.3

Ian Porteous, Arthur Asuncion, and Max Welling. Bayesian matrix factorization with side information and Dirichlet process mixtures. In *24th AAAI Conference on Artificial Intelligence*, 2010. 5.3

J. Scott Provan and Michael O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, 12(4):777–788, 1983. 4.2.2

Filippo Radicchi, Santo Fortunato, Benjamin Markines, and Alessandro Vespignani. Diffusion of scientific credits and the ranking of scientists. *Physical Review E*, 80:056103, 2009. 4.4

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2009.

5.2.2, 6.5

Daniel Ramage, Susan Dumais, and Dan Liebling. Characterizing microblogs with topic models. In *International Conference on Weblogs and Social Media (ICWSM)*, 2010. 5.3

Sidney Redner. How popular is your paper? an empirical study of the citation distribution. *European Physical Journal B*, 4: 131–134, 1998. 4.6

Nicholas D. Rizzolo and Dan Roth. Modeling discriminative global inference. In *Proceedings of the First International Conference on Semantic Computing (ICSC)*, 2007. 3.4

Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, 2005. 2.1.2

Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105:1118–1123, 2008. 4.6

Michal Rozen-Zvi, Thomas L. Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *20th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2004. 4.6

G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11), November 1975. 1.2, 3.1

Ross D. Schachter. Bayes-ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). In *14th Conference on Uncertainty in Artificial Intelligence (UAI)*, 1998. 2.1.1, 2.2

Benyah Shaparenko and Thorsten Joachims. Information genealogy: uncovering the flow of ideas in non-hyperlinked document databases. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2007. 4.5, 4.6

Ben Smith. The hair's still perfect. Politico, April 16, 2007. 3

Noah A. Smith. *Linguistic Structure Prediction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool, May 2011. 4

Alexander J. Smola and Shravan Narayanamurthy. An architecture for parallel topic models. *Proceedings of Very Large Data Bases (PVLDB)*, 3(1):703–710, 2010. 5.4

Victoria Stodden. *Model selection when the number of variables exceeds the number of observations*. PhD thesis, Stanford University, 2006. 2.2.1

Kazunari Sugiyama, Kenji Hatano, and Masatoshi Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *13th International World Wide Web Conference (WWW)*, 2004. 6.5

Romain Thibaux and Michael I. Jordan. Hierarchical beta processes and the Indian buffet process. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007. 5

Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58(1): 267–288, 1996. 2.2, 2.2.1, 2.2.2

Roberto Torres, Sean M. McNee, Mara Abel, Joseph A. Konstan, and John Riedl. Enhancing digital libraries with TechLens+. In *ACM/IEEE Joint Conference on Digital Libraries*, 2004. 4.6

Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979. 4.2.2

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. 2.1

Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. In *Exploring artificial intelligence in the new millennium*. Morgan Kaufmann, 2003. 2.1.2

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1):49–67, 2007. 2.2.1

Yisong Yue and Carlos Guestrin. Linear submodular bandits and their application to diversified retrieval. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS) 24*, 2011. 3.5, 6.1

Yisong Yue and Thorsten Joachims. Predicting diverse subsets using structural SVMs. In *25th International Conference on Machine Learning (ICML)*, 2008. 3.6.1

Jeffrey Zaslow. If TiVo thinks you are gay, here's how to set it straight. *The Wall Street Journal*, November 26 2002. 5, 5.3

Chengxiang Zhai, William W. Cohen, and John Lafferty. Beyond independent relevance: methods and metrics for subtopic

retrieval. In *26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2003. 3.6.1

Benyu Zhang, Hua Li, Yi Liu, Lei Ji, Wensi Xi, Weiguo Fan, Zheng Chen, and Wei-Ying Ma. Improving web search results using affinity graph. In *28th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval (SIGIR)*, 2005. 3.6.1