

UCLA

UCLA Electronic Theses and Dissertations

Title

Bayesian Modeling for Analyzing Online Content and Users

Permalink

<https://escholarship.org/uc/item/20k825jn>

Author

Bi, Bin

Publication Date

2015

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

**Bayesian Modeling for Analyzing Online
Content and Users**

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Bin Bi

2015

ABSTRACT OF THE DISSERTATION

Bayesian Modeling for Analyzing Online Content and Users

by

Bin Bi

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2015

Professor Junghoo Cho, Chair

The immense scale of the web has rendered itself as a huge content repository. Web users seek information content of interest primarily from search engines and social media. The sheer amount of online content, ranging from professionally-produced content to user-generated content, varies greatly in quality, which can often result in confusion, sub-optimum decisions or dissatisfaction with choices made by users. It is, therefore, highly significant to develop learning models that are able to automatically discover high-quality content for web users.

This thesis explores two general schemes toward this ultimate goal: 1. Learning to discover high-quality content and delivering it to users. 2. Learning to identify domain authorities who generate high-quality content, so users can obtain quality content from these authorities. Under the two schemes, we propose a range of Bayesian statistical models, each specifically designed for a unique application in social media or web search engines. These models are able to discover high-quality information by statistically analyzing the online content and users in the systems.

In particular, in the social media domain, we introduce a range of Bayesian models specifically designed to identify topic-specific influencers or experts in microblogs and content-sharing websites. On the other hand, in the search engine

domain, two different Bayesian models are proposed to analyze the search users and database. One of the models is specifically designed to build a recommender system on a knowledge base, which suggests related entities for search users, while the other model is developed to infer the demographics of users, which can be utilized to enhance their search experience. Extensive experiments have been conducted on real-world data to confirm the effectiveness of all the proposed models.

The dissertation of Bin Bi is approved.

Ying Nian Wu

Alfonso Cardenas

Carlo Zaniolo

Junghoo Cho, Committee Chair

University of California, Los Angeles

2015

TABLE OF CONTENTS

1	Introduction	1
1.1	Online Content and User Analysis	1
1.2	Social Media and Search Engine	2
1.3	Dissertation Outline	4
2	Background	5
2.1	Statistical Modeling	5
2.2	Latent Dirichlet Allocation	6
2.3	Dirichlet Process Mixture	9
2.4	Hierarchical Dirichlet Processes	11
3	Topic-specific Influence Analysis on Microblogs	13
3.1	Introduction	13
3.2	Related Work	16
3.3	Followship-LDA	18
3.3.1	Gibbs Sampling for FLDA	22
3.4	Scalable Gibbs Sampling for FLDA	23
3.4.1	Spark Overview	24
3.4.2	Distributed FLDA using Spark	26
3.4.3	Discussion	28
3.5	Querying Topical Influencers	29
3.6	Experiments	30
3.6.1	Effectiveness on Twitter Dataset	32

3.6.2	Effectiveness on Tencent Weibo Dataset	34
3.6.3	Scalability	37
3.7	Conclusion	38
4	Bayesian Nonparametric Modeling for Microblog Data Analysis	44
4.1	Introduction	44
4.2	User-Retweet Model (URM)	44
4.2.1	Generative Process for URM	46
4.2.2	URM as a Three-layer DP Hierarchy	48
4.2.3	Bayesian Inference for URM	49
4.3	User-centric Model (UCM)	52
4.3.1	Generative Process for UCM	52
4.3.2	Bayesian Inference for UCM	54
4.4	Empirical Evaluation	55
4.4.1	Dataset and Experiment Settings	55
4.4.2	Topics Produced by URM and UCM	56
4.4.3	Topic Quality	57
4.4.4	Predictive Power Analysis	60
4.4.5	Conclusion	61
5	Topic-specific Authority Analysis on Content Sharing Services	64
5.1	Introduction	64
5.2	Related Work	67
5.3	Problem Statement	69
5.4	Topic-specific Authority Analysis	73

5.5	Inference for TAA	77
5.5.1	Preference Learning	77
5.5.2	Bayesian Inference	79
5.5.3	Authority Analysis Framework	82
5.6	Empirical evaluation	83
5.6.1	Data Collections	83
5.6.2	Evaluation Strategy	84
5.6.3	Quality of Authority Analysis	85
5.6.4	Predictive Power Analysis	88
5.6.5	Case Visualization	89
5.7	Conclusion	90
6	Inferring the Demographics of Search Users	93
6.1	Introduction	93
6.2	Related Work	95
6.3	Modeling User Demographics	98
6.4	Data	103
6.5	Evaluation	104
6.6	Experiments	106
6.7	Importance of ODP categories	110
6.8	Conclusion	112
7	Learning to Recommend Related Entities to Search Users . . .	116
7.1	Introduction	116
7.2	Related Work	121

7.3	Problem Statement	122
7.4	Three-way Entity Model	127
7.4.1	Trilinear function	127
7.4.2	CTR incorporation	128
7.4.3	Likelihood function	129
7.4.4	TEM & Inference	132
7.4.5	Three-way Interaction Effect	135
7.5	Empirical evaluation	136
7.5.1	Data	136
7.5.2	Evaluation strategy	138
7.5.3	Recommendation accuracy	139
7.5.4	Efficacy of personalization	141
7.5.5	Effect of random projections	145
7.6	Conclusion	145
8	Conclusions and Future Work	147
	References	150

LIST OF FIGURES

2.1	Graphical model of LDA	7
2.2	Graphical model representation	9
3.1	Followship-LDA	18
3.2	Generative process for Followship-LDA	20
3.3	Average number of returned VIPs	35
3.4	Mean Average Precision	36
3.5	Speed-Up of Distributed FLDA on the Twitter dataset.	37
4.1	Graphical models for URM	48
4.2	Graphical models for UCM	52
4.3	Histogram of the number of latent topics produced during the Gibbs sampling process	62
4.4	Comparison of word perplexity for HDP, URM and UCM	63
5.1	Sample records from the sharing log of a photo sharing website	69
5.2	Sample records from the favorite log of a photo sharing website	72
5.3	Graphical model for Topic-specific Authority Analysis	75
5.4	Generative process for Topic-specific Authority Analysis	76
5.5	MRR for the Flickr dataset	86
5.6	Spearman’s rank correlation coefficient for the 500px dataset	88
5.7	Perplexity for the Flickr dataset	89
5.8	Perplexity for the 500px dataset	90

5.9	Examples of the ranked lists of photographers identified by <i>TAA</i> on Flickr data	91
6.1	The workflow of our framework for inferring users' demographics based on the search queries. On the left, the Facebook Likes of a small group of users are mapped to their corresponding ODP categories by issuing them as queries and classifying the top search results. On the right, the search users are represented similarly by the set of ODP categories associated with the top-ranked results returned for their queries.	100
6.2	<p>(Top-Left) The distribution of Christians in the <i>Contiguous</i> United States according to the U.S. Religious Landscape Survey. (Top-Right) The distribution of Christians in the US as predicted based on user queries. The Pearson correlation (ρ) is 0.39. (Bottom-Left) The distribution of Buddhism in the <i>Contiguous</i> United States according to the U.S. Religious Landscape Survey. (Bottom-Right) The distribution of Buddhism in the US as predicted based on user queries. The Pearson correlation (ρ) is 0.53. The spectrum bar at the left corner of each map specifies the scale and the corresponding color codes.</p>	113

6.3	(Top-Left) The distribution of Agnostics in the <i>Contiguous</i> United States according to the U.S. Religious Landscape Survey. (Top-Right) The distribution of Agnostics in the US as predicted based on user queries. The Pearson correlation (ρ) is 0.27. (Bottom-Left) The distribution of Judaism in the <i>Contiguous</i> United States according to the U.S. Religious Landscape Survey. (Bottom-Right) The distribution of Judaism in the US as predicted based on user queries. The Pearson correlation (ρ) is 0.54. The spectrum bar at the left corner of each map specifies the scale and the corresponding color codes.	114
6.4	(Top) The outcome of 2012 the US presidential election according to The Huffington Post. The blue states were won by Democrats and the red states by Republicans. (Bottom-Left) The distribution of conservatives versus liberals according to an independent poll – Gallup. (Bottom-Right) Liberal vs. conservative predictions on Bing users based on the models learned according to Facebook data. The Pearson correlation (ρ) between the Gallup data and our per-state predictions is 0.72. The spectrum bar at the left corner of each map specifies the scale and the corresponding color codes.	115
7.1	Example of the entities and their relations taken from Freebase . . .	118
7.2	Example of search results with the entity pane taken from a commercial web search engine	119
7.3	Sample records taken from an entity pane log	122
7.4	Illustration on joining search click log with Freebase knowledge base	125
7.5	Preference relations induced by the click feedback in the entity pane log	131

7.6	Graphical representation of Topic-specific Authority Analysis . . .	133
7.7	MRR for movie recommendation	141
7.8	RankAcc for movie recommendation	142
7.9	MRR for celebrity recommendation	143
7.10	RankAcc for celebrity recommendation	143
7.11	User distribution and MRR	145
7.12	MRR for varying random projection dimensions	146

LIST OF TABLES

3.1	Notations used in FLDA	19
3.2	Statistics of Experimental Datasets.	31
3.3	A sample of FLDA topics and their influencers.	31
3.4	A sample of the topics of FLDA and Link-LDA with their influencers	32
4.1	A sample of latent topics produced by URM and UCM	57
4.2	Comparison of model precisions	59
4.3	Model precisions of URM and UCM over tweet/retweet topics . .	59
5.1	Notations used throughout this chapter	71
5.2	Statistics of Experimental Datasets	84
6.1	The distribution of age and gender in search queries and Facebook Likes datasets.	103
6.2	The Area under the ROC Curve (AUC) for different demographic prediction models. The numbers in the middle column show the AUC of a model trained on Facebook data for predicting the de- mographics of Facebook users. In the right column, the models trained based on Facebook data are tested on search query sample. The missing values “-” are used where the per-user ground-truth information is not available for AUC evaluation.	106
6.3	The ODP categories with the highest information gain for different types of demographics.	110
7.1	Notations used throughout this chapter	124
7.2	Statistics of experimental datasets	136

7.3	Features for movie & celebrity recommendation	138
7.4	Related movies recommended for a fan of actor Leonardo DiCaprio by <i>Co-click</i> , <i>Production</i> , <i>CTR-model</i> , and <i>TEM</i>	144

ACKNOWLEDGMENTS

This work would not have been possible without the support of my faculty advisor, academic collaborators, family and friends.

First, I would like to extend my deepest gratitude to my faculty advisor, Professor John Cho. Every time I discussed with him, I would benefit from his experience on the way of doing research, his deep insight on the research topics, and his advices on furthering our research and making effective writings and presentations. John offered me unending support, inspiring me to play with new ideas and encouraging me to explore the direction which I am interested in, for which I am truly thankful.

I would also like to include my gratitude to all my colleagues in Microsoft Research Redmond, Microsoft Research Cambridge and IBM Almaden Research Center for allowing me to gain valuable industrial experience. I really appreciate their insightful guidance and advices towards completion of my research work during my internships with their labs.

Last but not least, I would like to thank my family and friends for their limitless love and support. I was fortunate to have parents who loved me unconditionally and supported me bottomlessly. This thesis is dedicated to my family, without which this work would not have been possible. Also, I want to say thanks to all my friends for their moral support during my Ph.D. studies. They all helped me make my life rich and colorful.

VITA

- 2004–2008 B.E. (Software Engineering), Huazhong University of Science and Technology, Wuhan, China
- 2008–2010 M.Phil. (Computer Science), The University of Hong Kong, Hong Kong
- 2010–2015 Doctoral Student (Computer Science), University of California, Los Angeles, California, USA

PUBLICATIONS

- **Bin Bi**, Hao Ma, Paul Hsu, Wei Chu, Kuansan Wang, and Junghoo Cho. *Learning to Recommend Related Entities to Search Users*. Proceedings of the 8th ACM International Conference on Web Search and Data Mining (**WSDM**), 2015.
- **Bin Bi**, Ben Kao, Chang Wan, and Junghoo Cho. *Who Are Experts Specializing in Landscape Photography? - Analyzing Topic-specific Authority on Content Sharing Services*. Proceedings of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (**KDD**), 2014.
- **Bin Bi**, Yuanyuan Tian, Yannis Sismanis, Andrey Balmin, and Junghoo Cho. *Scalable Topic-Specific Influence Analysis on Microblogs*. Proceedings of the 7th ACM International Conference on Web Search and Data Mining (**WSDM**), 2014.

- **Bin Bi**, Milad Shokouhi, Michal Kosinski, and Thore Graepel. *Inferring the Demographics of Search Users - Social Data Meets Search Queries*. Proceedings of the 22nd International World Wide Web Conference (**WWW**), 2013.
- **Bin Bi**, and Junghoo Cho. *Automatically Generating Descriptions for Resources by Tag Modeling*. Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (**CIKM**), 2013.
- Youngchul Cha, **Bin Bi**, Chu-Cheng Hsieh, and Junghoo Cho. *Incorporating Popularity in Topic Models for Social Network Analysis*. Proceedings of the 36th International ACM SIGIR Conference (**SIGIR**), 2013 (**Best Paper Runner Up**).

CHAPTER 1

Introduction

1.1 Online Content and User Analysis

We have entered the era of big data, which is characterized by an explosion of information content. Data sources are everywhere online, from professionally-produced content to Web 2.0 and user-generated content, from news sites to social media. Based on the infographic¹ from analytics software provider Domo, nowadays a massive amount of data is generated every minute on the Internet. For example, content sharing websites, such as YouTube² and Vine³, have become tremendously popular over the recent years. In one minute, there is a staggering 72 hours of content uploaded to YouTube, while Vine users share 8,333 videos. Searching Google is still a most-popular activity online with more than four million search queries per minute. When it comes to social networks, Facebook holds dominion with users posting nearly 2.5 million pieces of content, while Twitter users create 277 thousand new tweets.

The sheer amount of online content available can be both a blessing and a curse for web users. On the plus side, it is a valuable asset in our information society. The massive amount of data enables users to investigate content on topics of their personal interest, to read up on what is happening in the world, and to get advice about problems and things they don't much like to talk about publicly, etc. On

¹http://web-assets.domo.com/blog/wp-content/uploads/2014/04/DataNeverSleeps_2.0_v2.jpg

²<http://www.youtube.com>

³<http://www.vine.co>

the minus side, however, the huge amount of online content can often complicate the decision making process, since users don't have the time or ability to examine all data or compare all options. Not only is the volume of content overwhelming, but also the quality of the content can be far from perfect. As we know, online content varies greatly in quality, which results in confusion, sub-optimum decisions or dissatisfaction with the choices made by users.

Given that online content remains confusing, difficult to consolidate, and often chaotic, it is highly significant to develop learning models that are able to discover high-quality content for web users. There are two different schemes toward the ultimate goal. The first scheme is discerning good content from the bad. In other words, we are sorting the wheat from the chaff. Given a huge diversity of online content, appropriate modeling of the content plays an important role in discovering high-quality content relevant to users' interest. The other scheme toward our ultimate goal is identifying experts or key influencers who generate high-quality content. These experts act as gatekeepers who pose certain tests of quality and authenticity. The content that passes these tests is supposed to be high-quality and authentic. These experts consequently serve as mediators that web users can obtain high-quality content from. Therefore, in this dissertation, we present a family of statistical models specifically designed for understanding and analyzing online content and users under the two schemes, in order to discover high-quality and relevant content for web users.

1.2 Social Media and Search Engine

Traditionally, on the web, people seek information content of their interest primarily from web search engines, such as Google and Bing. However, the recent rise

in popularity of social media, such as Facebook⁴, Twitter⁵ and Flickr⁶, has introduced a new option for finding online content of interest. To analyze the content and users in social media and search engines, we develop different probabilistic models, each for a specific class of applications in these two domains.

Over the last few years, we have been witnessing the rapid emergence of various facets of social media on the web, including blogs, content-sharing websites, social tagging systems, social networking platforms, and microblogs, which has provided a vast continuous supply of dynamic diverse information content. Andreas Kaplan and Michael Haenlein define social media as “a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of user-generated content.” [65]. When analyzed with appropriate statistical and computational tools, social media content can be turned into invaluable scientific and business insights. In this dissertation, we introduce a range of Bayesian models to analyze the content and users in microblogs and content-sharing websites, which are different types of social media [20, 15, 14].

In addition to social media, web search engines are a important tool for online information seeking [113]. Proper understanding and modeling of the data residing in a search engine allow a search user to discover high-quality content relevant to his or her unique information needs. For example, with the introduction of *knowledge bases* over the past few years, commercial search engines are moving towards retrieval based on semantic understanding of search queries. A knowledge base is a centralized repository of content about entities including people, places and things. Modeling and utilizing the knowledge base properly enables the search engine to provide structured and detailed information about the query topic. It allows the search users to use this information to resolve their

⁴<http://www.facebook.com>

⁵<http://www.twitter.com>

⁶<http://www.flickr.com>

query without having to assemble information from other sites themselves. In this dissertation, we present two statistical models to analyze the content and users in search engines, respectively. One of the models is devised to build a recommender system on a knowledge base, which suggests related entities for search users [17], while the other model is specifically designed to infer the demographics of users, which can be leveraged to enhance their search experience [19].

1.3 Dissertation Outline

The remainder of the dissertation is organized in the following manner:

Chapter 2 reviews the theoretical background on statistical modeling and a couple of specific Bayesian models, which are closely related to our proposed models presented in the following chapters. In Chapter 3, we introduce the FLDA model, which is an extension of the typical topic model, specifically designed for topic-specific social influence analysis on microblogs, a popular form of social media. By contrast, Chapter 4 presents different models based on the Bayesian nonparametric framework to analyze microblog data. In Chapter 5, we describe a different Bayesian model, TAA, specifically designed to discover topic-specific authorities on content-sharing websites, another form of social media.

In addition to social media, we present our work on analyzing data residing in web search engines in Chapters 6 and 7. In particular, Chapter 6 introduces predictive models to infer the demographics of search users, while in Chapter 7 we describe a novel recommender system that suggests related entities on the knowledge pane of a search engine. Finally, Chapter 8 concludes the dissertation and discusses directions for future research work.

CHAPTER 2

Background

2.1 Statistical Modeling

Mathematically, a statistical model is a set of assumptions concerning the generation of the observed data from a larger population. The model is formally specified by relationships among one or more random variables and other variables. Statistical modeling has two complementary and important paradigms: discriminative models and generative models. Discriminative models are designed to capture decision boundaries among different classes. Discriminative modeling makes a strong assumption that class labels are available in data, but it has the virtue that few additional assumptions are generally needed to build a useful model. On the other hand, generative modeling, which does not require class labels, probabilistically expresses hypotheses about the way in which observed data may have been generated. In our work, we choose one class of models over the other based on the distinct nature of problems and applications.

Another important dichotomy in statistical modeling distinguishes between parametric modeling and nonparametric modeling. A parametric model fixedly explains observed data in a way that it does not grow structurally as more data come. Examples include Latent Dirichlet Allocation (LDA) [23] and Gaussian Mixture Models (GMMs) where the number of latent topics (clusters) has to be determined a priori and remains fixed throughout the models. A nonparametric model, on the other hand, allows the representation of data to grow structurally

as more data are observed. As opposed to a parametric model, it is capable of letting the data speak for itself to automatically determine the complexity of the nonparametric model. The Dirichlet Process Mixture (DPM) model [6] is the key building block in Bayesian nonparametric models for a broad range of applications. The DPM model has been extended to Hierarchical Dirichlet Processes (HDP) [101] to cluster grouped data.

For the purpose of clarity, later in this chapter, we briefly review a parametric topic model, LDA, and two typical nonparametric models, DPM and HDP, which are the main constituents of our proposed models described in the following chapters.

2.2 Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) [23] is a generative topic model which has attracted a lot of interest from both the machine learning and language processing community. It is essentially the Bayesian version of the PLSA model [56]. In general, a Bayesian treatment of a statistical model enables a fully probabilistic (i.e., Bayesian) approach to learning the model, which updates a prior model into a posterior model once data have been observed.

In the LDA model, documents are represented as random mixtures over topics, where each topic is characterized by a distribution over words. More precisely, each individual word token w_n in a corpus $\mathbf{w} = \{w_1, w_2, \dots, w_N\}$ is assumed to have been derived from a single topic z_n , drawn from a document-specific distribution over K topics. The probability of generating a word w from a topic k is defined by $\phi_{w|k} = p(w_n = w | z_n = k)$. These probabilities are recorded in a $K \times V$ matrix Φ , where K is the number of topics and V is the size of vocabulary. Similarly, the topic generation is characterized by the probability $\theta_{k|d} = p(z_n = k | d_n = d)$. These probabilities are recorded in a $M \times K$ matrix Θ , where M is the number

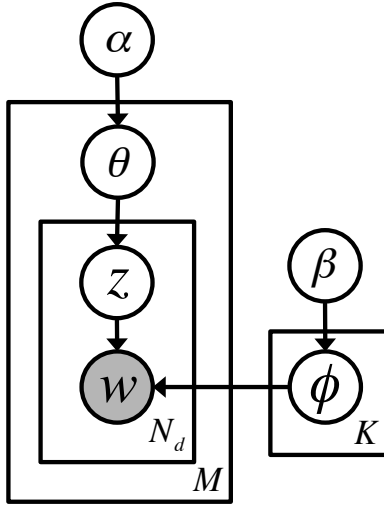


Figure 2.1: Graphical model of LDA

of documents in the corpus. Thus the joint probability of corpus \mathbf{w} and a set of corresponding topics $\mathbf{z} = \{z_1, \dots, z_N\}$ is:

$$p(\mathbf{w}, \mathbf{z} | \Phi, \Theta) = \prod_{n=1}^N \phi_{w_n | z_n} \theta_{z_n | d_n}, \quad (2.1)$$

where w_n is the n -th word of the corpus \mathbf{w} , z_n is the topic assignment for the n -th word and d_n denotes the document of the n -th word.

In order to make the model fully Bayesian, symmetric Dirichlet priors with hyperparameters α and β are placed over Θ and Φ , i.e.,

$$p(\Theta | \alpha) = \prod_d \text{Dirichlet}(\theta_d | \alpha) \quad (2.2)$$

$$p(\Phi | \beta) = \prod_k \text{Dirichlet}(\phi_k | \beta), \quad (2.3)$$

where θ_d is the d -th row of the matrix Θ , ϕ_k is the k -th row of the matrix Φ . Incorporating the two priors into Equation (5.1) and integrating over Θ and Φ gives $p(\mathbf{w}, \mathbf{z} | \alpha, \beta)$, the joint probability of corpus and topics given hyperparameters. The conditional dependencies implied by this joint distribution can be represented by the graphical model shown in Figure 2.1. The repeated generation of topics and words can be illustrated by the plate notation with the number in the

right-lower corner indicating the number of repetitions. The shaded and unshaded circles represent observed and latent (i.e, unobserved) variables, respectively. The arrows indicate conditional dependencies between the random variables.

As a result, the posterior probability for the topic assignments specified by latent variables \mathbf{z} is given by:

$$p(\mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{p(\mathbf{w}, \mathbf{z}|\alpha, \beta)}{\sum_{\mathbf{z}} p(\mathbf{w}, \mathbf{z}|\alpha, \beta)}. \quad (2.4)$$

Unfortunately, exact inference of the LDA model is intractable, since computing Equation (2.4) involves evaluating a probability distribution on a large discrete state space. However a number of methods of approximating the posterior distribution have been proposed including mean field variational inference [23] and Gibbs sampling [52, 95]. Gibbs sampling is a Markov chain Monte Carlo method where a Markov chain is constructed that slowly converges to the target distribution of interest over a number of iterations. For LDA each Gibbs sample from the posterior distribution (2.4) is obtained by:

$$p(z_n = k|z_{-n}, \mathbf{w}, \alpha, \beta) \propto \frac{N_{-n,k}^{(w_n)} + \beta}{N_{-n,k}^{(\cdot)} + V\beta} \frac{N_{-n,k}^{(d_n)} + \alpha}{N_{-n}^{(d_n)} + K\alpha} \quad (2.5)$$

where \mathbf{z}_{-n} denotes all the z_j with $j \neq n$, $N_{-n,k}^{(w_n)}$ is the number of times the word w_n is assigned to topic k and $N_{-n,k}^{(\cdot)}$ is the number of words assigned to topic k . $N_{-n,k}^{(d_n)}$ is the number of times topic k occurs in document d_n and $N_{-n}^{(d_n)}$ is the number of words in document d_n . All the four counts exclude the current assignment of z_n . After the sampling algorithm has been run over each word position in the corpus an appropriate number of times (i.e., until the chain has converged to a stationary distribution) we sample from the distribution to obtain estimates for our parameters Φ and Θ via the following equations:

$$p(w|k) = \phi_{w|k} = \frac{N_k^{(w)} + \beta}{N_k^{(\cdot)} + V\beta} \quad (2.6)$$

$$p(k|d) = \theta_{k|d} = \frac{N_k^{(d)} + \alpha}{N^{(d)} + K\alpha} \quad (2.7)$$

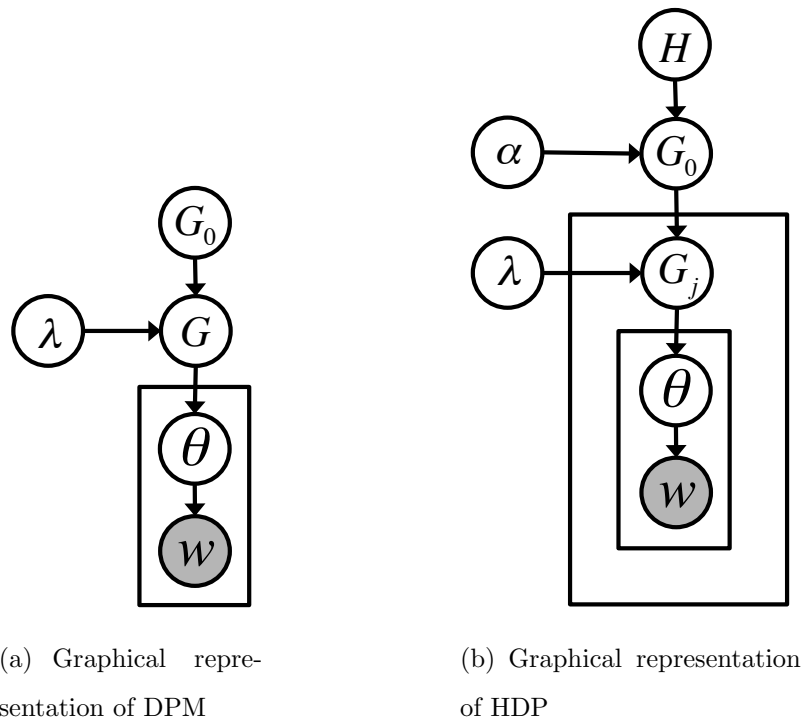


Figure 2.2: Graphical model representation

2.3 Dirichlet Process Mixture

LDA requires the number of topics as an input parameter which remains fixed throughout the model. Setting the number of topics is a perennial question in topic modeling. One way of sidestepping this issue is applying ideas from Bayesian nonparametric statistics, in which the Dirichlet Process Mixture (DPM) model [6] is the key building block for a broad range of applications. There are three different views on the DPM model: (1) a distribution of a random probability measure, (2) intuitive Chinese Restaurant Process (CRP), and (3) a limit of a finite mixture model. All of these perspectives are equivalent, but each one provides a different view of the same process, and some of them might be easier to follow.

A *Dirichlet process* (DP) is defined as a distribution of a random probability measure G [42]. A DP, denoted by $DP(\lambda, G_0)$, is parameterized by a concentration parameter λ , and a base measure G_0 . $G \sim DP(\lambda, G_0)$ denotes a draw of a random

probability measure G from the Dirichlet process. G is technically a distribution over a given parameter space θ , so one can draw parameters $\theta_1, \dots, \theta_n$ from G . Previously drawn values of θ_i have strictly positive probability of being redrawn again, which makes the underlying probability measure G discrete [21]. Using a DP at the top of a hierarchical model leads to the Dirichlet Process Mixture model for Bayesian nonparametric modeling [6].

Figure 2.2(a) depicts the graphical model representation of a DPM model. Formally, sampling from a DPM model is conducted by the following generative process:

$$\begin{aligned} G &\sim DP(\lambda, G_0), \\ \theta_i &\sim G, \\ w_i &\sim F(\cdot|\theta_i) \end{aligned} \tag{2.8}$$

where F is a given likelihood function parameterized by θ . The clustering property of a DP prefers to use fewer than n distinct θ . An equivalent Chinese Restaurant Process metaphor exhibits the clustering property. In particular, consider a Chinese restaurant with an unbounded number of tables. Each θ_i corresponds to a customer who enters the restaurant. The i -th customer θ_i sits at table k that already has n_k customers with probability $\frac{n_k}{i-1+\lambda}$, and shares the dish (parameter) ψ_k served there, or sits at a new table with probability $\frac{\lambda}{i-1+\lambda}$, and orders a new dish sampled from G_0 . This process can be expressed as:

$$\theta_i|\theta_1, \dots, \theta_{i-1}, \lambda, G_0 \sim \sum_{k=1}^{i-1} \frac{n_k}{i-1+\lambda} \delta_{\psi_k} + \frac{\lambda}{i-1+\lambda} G_0. \tag{2.9}$$

where δ_ψ is a probability measure concentrated at ψ .

Finally, a DPM model can be derived as the limit of a sequence of finite mixture models, where the number of mixture components is taken to infinity. Therefore, a DPM can be used to build an infinite-dimensional mixture model, and has the desirable property of extending the number of clusters with the arrival of new data. This flexibility enables the DPM to conduct model selection automatically.

2.4 Hierarchical Dirichlet Processes

The DPM is widely used to build a model with a discrete random variable of unknown cardinality (i.e., a cluster indicator). The HDP, on the other hand, applies to the problems in which multiple different groups of data would share the same settings of partitions. In such applications, the model for each of the groups incorporates a discrete variable of unknown cardinality. The HDP model is able to share clusters across multiple clustering problems.

The key building block of the HDP model is a recursion where the base measure G_0 for a DP: $G \sim DP(\lambda, G_0)$ is itself a draw from another DP: $G_0 \sim DP(\alpha, H)$. By this recursive construction, the random measure G are constrained to place its atoms at the discrete locations determined by G_0 . Such a construction is commonly used for conditionally independent hierarchical models of grouped data.

More formally, in HDP, we model each of the groups as a DP, which is gathered into an indexed collection of DPs $\{G_j\}$. In order to be tied probabilistically, the random measures share their base measure, which is defined to be random as well, as follows:

$$\begin{aligned} G_0 &\sim DP(\alpha, H) \\ G_j &\sim DP(\lambda, G_0). \end{aligned} \tag{2.10}$$

This means that we first draw G_0 from the base measure H . The random measure G_0 is then, in turn, used as a reference measure to obtain the measures G_j . As a result, each random measure G_j inherits its set of atoms from the same G_0 . Therefore, this conditionally independent hierarchical model induces sharing of atoms among these random measures G_j . The graphical model of HDP is shown in Figure 2.2(b).

Integrating out all random measures, we obtain the equivalent Chinese Restaurant Franchise processes (CRF) [101]. In the CRF, the metaphor of a Chinese

restaurant is extended to a set of restaurants which share a set of dishes. The customers in the j -th restaurant sit at tables in the same manner as the CRP, and this is done independently in the restaurants. The coupling among restaurants is achieved via a franchise-wise menu. The first customer to sit at a table in a restaurant chooses a dish from the menu and all subsequent customers who sit at that table inherit that dish. Dishes are chosen with probability proportional to the number of tables (franchise-wide) which have previously served that dish.

CHAPTER 3

Topic-specific Influence Analysis on Microblogs

3.1 Introduction

Microblogging services, such as Twitter (`twitter.com`), have gained tremendous popularity in recent years. Using these services, a user can publish a short message, called a *tweet*, and *follow* other users to keep up with their latest updates. The “follow” relationship (or *followership*) is directed, with information only flowing from the *followee* to the *follower*. A large amount of microblog data has been accumulated over time. For example, according to a March 2012 report, Twitter had over 500 million registered users creating over 340 million tweets daily [104].

The rich text and social information in microblogs has become a popular resource for marketing campaigns to monitor the opinions of consumers on particular products and to launch viral advertising. Identifying key influencers in microblogs is required for such marketing activities. Although a lot of work has been done on social influence analysis, most of these studies [29, 62, 67, 72, 33] infer influence only from the network structure, while ignoring the valuable text content that the users created. As a result, the learned influence of each user is only *global*, with no way to assess the influence in a particular aspect of life (topic). For example, no one can deny that President Obama is a key influencer in general. But his impact is most prominent in politics. In other subjects, like choosing digital cameras, he is unlikely to be influential. Clearly, *topic-specific* influence analysis provides a more detailed influence portfolio for a user, which is critical for effective

marketing.

A number of PageRank-based methods, such as Topic-Sensitive PageRank [54] and TwitterRank [114], are able to compute per-topic influence ranks, but they require the topics to be already created either manually or by a topic modeling preprocess. As content and links are related to each other in a microblog network, the separation between the analysis on content and the analysis on the network structure usually leads to inferior performance, compared to those methods, like Link-LDA [39], which can detect topics and infer influences at the same time. However, Link-LDA, as originally designed for citation networks, assumes that the generation of links is purely based on the content. This assumption clearly does not apply to microblogs, since it is prevalent for a user to follow celebrities simply because of their fame and stardom, with nothing to do with what he/she actually tweets about.

To correctly model topic-specific influence on microblogs, we propose a new Bernoulli-Multinomial mixture model, called Followship-LDA (FLDA). This model contains two levels of mixtures: an upper-level Bernoulli mixture with one of the components being a Multinomial mixture. FLDA jointly models text and followship in the same generative process. Furthermore, it is able to differentiate the different reasons why a user follows another. Sometimes, A follows B because they tweet in similar topics. This type of followship is content-based. In other times, A follows B purely because B is a pop star. In this case, the followship is content-independent. Using FLDA, we can not only learn the per-user preference of following by content or not, but also remove the stardom effect when computing the topic-specific influence. Our empirical study on two popular microblog datasets, Twitter and Tencent Weibo, shows that the FLDA model produces significantly higher quality results than the prior arts.

Gibbs sampling is a widely used approach to approximate target distributions for LDA-like Bayesian models. To meet the computational challenge posed

by rapid growing microblog data, we propose a *distributed* Gibbs sampling algorithm which significantly speeds up the Gibbs sampling process. For example, a sequential job that would take 21 days on a high end server can finish in 1.5 days using the distributed algorithm on a cluster of 27 commodity machines! We chose to implement the distributed Gibbs sampling on top of the Spark cluster computing framework [118]. Several alternative platforms [26, 10, 24] have been proposed to address the problem of machine-learning at scale. Spark is such a parallel programming framework, which supports efficient iterative algorithms on datasets stored in the aggregate memory of a cluster. We pick Spark as the underlying framework, because of its extreme flexibility as far as cluster programming is concerned. In addition to machine-learning algorithms, which were the main motivation behind the design of Spark, various data-parallel applications can be expressed and executed efficiently using Spark; examples include MapReduce, Pregel[75], HaLoop[26] and many others(see [118]).

Finally, we propose a general search framework for topic-specific key influencers, which can flexibly plug in various topic-specific influence methods, including FLDA, Link-LDA, Topic-Sensitive PageRank and TwitterRank. A user just needs to enter a set of keywords to describe his/her interest, the search framework will infer a topic distribution from the keywords and return a ranked list of influencers in the corresponding topic combination.

In particular, this work makes the following contributions:

- We propose a new Bayesian Bernoulli-Multinomial mixture model, FLDA, to jointly model both content and links in the same generative process, while separating the various reasons why a user follows another in a microblog network.
- We discuss and implement a distributed Gibbs-sampling technique for training FLDA over large clusters.

- We propose a general search framework for finding topic-specific key influencers with various models (including FLDA, Link-LDA and PageRank variants).
- Through extensive experimentation with two large real datasets, we demonstrate (a) the substantial better precision achieved by FLDA than previous work, (b) the excellent scalability of the distributed Gibbs-sampling technique over large clusters and (c) various interesting insights gained from the real datasets.

3.2 Related Work

Much work has been done on influence analysis in social networks. Kempe et. al. pioneered the Linear Threshold Model and Independent Cascade Model to explain the spread of influence in a social network and abstracted the key influencer problem into a maximization problem [67]. Along with subsequent works, such as [72] and [33], these methods are only after the identification of *global* influencers instead of influencers for specific topics. Although Barbieri et. al. extended the Linear Threshold Model and Independent Cascade Model to be topic-aware [9], the topics are still obtained based on the network structure, while totally ignoring the valuable content information.

Given the popularity of PageRank [25], it is only natural to extend it for topical influence analysis. Topic-Sensitive PageRank (TSPR) [54] was such an extension for computing per-topic PageRank scores. TSPR biases the computation of PageRank by replacing the classic PageRank’s uniform teleport vector with topic-specific ones. However, it requires a separate preprocess to create topics and provide per-topic teleport vectors. This preprocess can be done by either utilizing existing manually categorized topic hierarchies, such as suggested in [54], or applying well-known topic modeling methods like LDA [23] on the text content,

as suggested in [114].

In [114], TwitterRank was proposed to find topic-level influencers on Twitter. A set of topics is first produced by LDA on the tweets. Then TwitterRank applies a method similar to TSPR to compute the per-topic influence rank. The transition probability between two users in TwitterRank is defined based on the number of tweets published by different followees and the topical similarity between the follower and the followee.

In the context of documents and citations (or hyperlinks), a mixed membership model was proposed in [39] to jointly model text and citations in the same generative process, which we will refer to as Link-LDA. In the generative process of Link-LDA, for a given document, a citation to another document is created in exactly the same way as a word is created, and they share the same per-document topic distribution. If we aggregate all the tweets for each user, and treat a user as a document and his/her followships to other users as citations, then Link-LDA can be applied to the microblog network, to learn the probability of each microblog user u being followed by someone given a specific topic t . This probability can be used to measure u 's influence on the topic t . Link-PLSA-LDA [79] is an extension to Link-LDA, but also assumes that the cause of links is purely based on the text content.

In [100], Tang et al. proposed a Topical Affinity Propagation (TAP) model for topic-level social influence. But, similar to TSPR and TwitterRank, TAP requires a separate topic modeling approach to be applied first to derive a set of topics on the content.

In [86], Pal et. al. identify topical authorities by clustering users using 15 carefully selected features and then rank users within each cluster. Cognos [45] heavily relies on the manually curated Twitter "Lists" to infer topics of expertise and rank experts for different topics. Liu et al. introduced a graphical model to learn influence in the context of general heterogeneous networks [73].

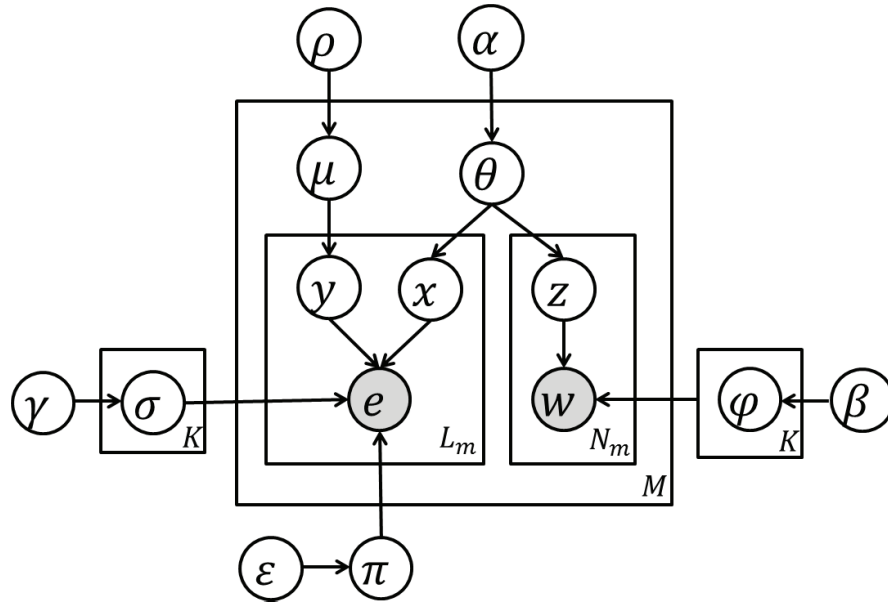


Figure 3.1: Followship-LDA

3.3 Followship-LDA

The existing works on topic-specific influence analysis can be categorized into two camps. The first camp, represented by TSPR and TwitterRank, completely detach the topic detection process from the influence analysis. As we will show later in this chapter, these methods perform inferior to those approaches in the second camp that integrate text topic discovery and social influence analysis in the same model. Link-LDA represents the best prior work in the second camp. However, it was originally developed for citation and hyperlink networks. In Link-LDA, the topic assignments for words and for citations are drawn from the same topic distribution θ , assuming that the content of a document is topically related to that of its cited documents. This is a very reasonable assumption for citation/hyperlink networks, since an author most definitely chooses the topically related documents to cite. But this assumption no longer applies to microblog networks. There are many reasons for a microblog user to follow another. Some are content-related (they tweet in similar topics) and others are not. For example,

Table 3.1: Notations used in FLDA

Notation	Description
θ	Per-user topic distribution
φ	Per-topic word distribution
σ	Per-topic followee distribution
π	Multinomial distribution over followees
μ	Per-user Bernoulli distribution over indicators
$\alpha, \beta, \gamma, \epsilon, \rho$	Parameters of the Dirichlet (Beta) priors on Multinomial (Bernoulli) distributions
w	Word identity
e	Followee identity
z	Identity of the topic of a word
x	Identity of the topic of a followee
y	Binary indicator of whether a followship is related to the content of tweets
M	Number of unique users
V	Number of words in the vocabulary
K	Number of unique topics
N_m	Number of words in the tweets of user m
L_m	Number of followees for user m

President Obama has a massive number of followers in Twitter, but some of them have never tweeted about politics at all. It is very common to see users follow celebrities, not because they share any topic of interest, but just because they are famous and popular. Clearly, Link-LDA is not able to capture these non-content related factors in the influence analysis.

To correctly model the topics and social influence in microblog networks, we propose **Followship-LDA**, abbreviated as FLDA. The graphical model for FLDA is depicted in Figure 3.1, with the notations described in Table 7.1. The generative process of a user’s content and links/followees is summarized in Figure 3.2.

For the generation of content, each user is viewed as a mixture of latent topics

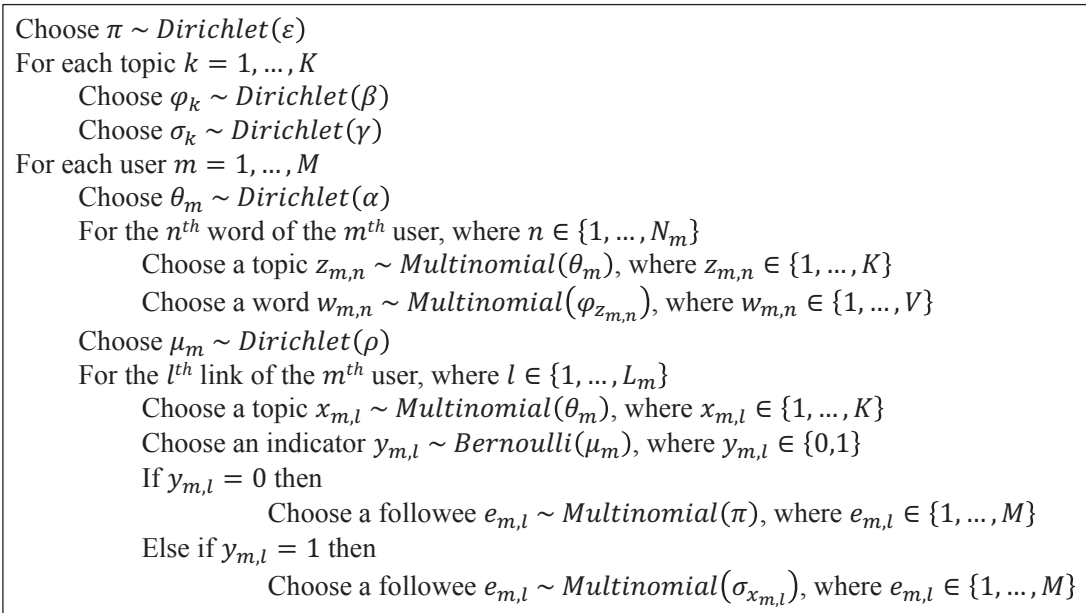


Figure 3.2: Generative process for Followship-LDA

from which words are drawn, similar to LDA. To be more specific, for the m^{th} user, we first pick the per-user topic distribution θ_m from a Dirichlet prior with parameter α . Then, to generate the n^{th} word for the tweets of the user, a topic $z_{m,n}$ is first chosen from θ_m . Finally, the word $w_{m,n}$ is picked from the per-topic word distribution $\varphi_{z_{m,n}}$.

On the other hand, the links of the m^{th} user are generated by a much more complex three-stage stochastic process. First of all, every user has a unique preference of following others based on content or non-content reasons. The Bernoulli distribution μ_m characterizes this per-user preference. As a result, for the l^{th} link/followee of the m^{th} user, we first consult μ_m to decide on the value of the binary variable $y_{m,l}$. $y_{m,l} = 1$ indicates that the link creation is based on the user's content, whereas $y_{m,l} = 0$ means that content has nothing to do with the link. Now if $y_{m,l} = 1$, we use the same topic distribution θ_m to pick a topic $x_{m,n}$ of interest, just as in the content generation part of FLDA. Afterward, we choose a followee $e_{m,l}$ who well addresses the picked topic from the per-topic followee distribution

$\sigma_{x_{m,l}}$. When $y_{m,l} = 0$, the user is following someone for non-content reasons. We use π to capture this probability distribution. In particular, a followee $e_{m,l}$ is chosen from the Multinomial distribution π .

Note that FLDA is a much more complex mixture model than LDA and Link-LDA. We call it a Bernoulli-Multinomial mixture model, because the model consists of two levels of mixtures: an upper-level Bernoulli mixture that includes a Multinomial mixture underneath. More specifically, each followee e of a user m is drawn from a Bernoulli mixture of two components. One of the mixture components is a Multinomial distribution with parameter π , corresponding to the global popularity. The other component, however, is itself a mixture of K Multinomial components, each corresponding to a topic. The distribution of followee e of user m is: $p(e|\mu, \pi, \theta, \sigma) = \mu_{m,0}\pi_e + \mu_{m,1} \sum_{k=1}^K \theta_{m,k}\sigma_{k,e}$, where μ are the outer mixing proportions, and θ are the inner mixing proportions.

The various probability distributions we can learn from the FLDA model characterize the different factors that affect the textual and social structures of a microblog network. For a user m , the probability $\theta_{z|m}$ represents the likelihood of m tweeting about topic z , and $\mu_{y|m}$ is the probability of the reason indicator y (content-related or not) why the user m follows others. For content of tweets, $\varphi_{w|z}$ gives the probability of word w belonging to topic z . In terms of links, $\sigma_{e|x}$ captures the likelihood of a user e being followed by someone for a given topic x . This value essentially quantifies the influence of user e on x and is exactly the topic-specific influence score we want to compute. Finally, π_e indicates the probability of a user e being followed for any non-content reason. In some sense, π_e is measuring the global popularity of e . We formally define:

Topic-Specific Influence: the influence of user e on topic x is measured by $\sigma_{e|x}$ which is the probability of e being followed for topic x in the FLDA model.

Content-Independent Popularity: the content-independent popularity of user e is measured by π_e which is the probability of e being followed for any content-

independent reason in the FLDA model.

3.3.1 Gibbs Sampling for FLDA

To learn the various distributions in the FLDA model, we use collapsed Gibbs sampling. However, the derivation of posterior distributions for Gibbs sampling in FLDA is complicated by the fact that followee distribution is a joint distribution of two-level mixtures. As a result, we need to compute the joint distribution of x and y in the Gibbs sampling process. The posterior distributions for Gibbs sampling in FLDA are given in the equations below. The detailed derivation of these equations is provided in the appendix.

$$\begin{aligned}
& p(z_{m,n} | z_{-(m,n)}, x, w, e, y, \alpha, \beta, \gamma, \varepsilon, \rho) \\
\propto & \frac{(c_{z_{m,n}, m, *}^{-(m,n)} + d_{z_{m,n}, m, *, *} + \alpha_{z_{m,n}})(c_{z_{m,n}, *, w_{m,n}}^{-(m,n)} + \beta_{w_{m,n}})}{c_{z_{m,n}, *, *}^{-(m,n)} + \sum_{i=1}^W \beta_i} \quad (3.1)
\end{aligned}$$

$$\begin{aligned}
& p(x_{m,l}, y_{m,l} = 0 | y_{-(m,l)}, x_{-(m,l)}, w, z, e, \alpha, \beta, \gamma, \varepsilon, \rho) \\
\propto & (c_{x_{m,l}, m, *} + d_{x_{m,l}, m, *, *}^{-(m,l)} + \alpha_{x_{m,l}})(d_{*, m, *, 0}^{-(m,l)} + \rho_0) \times \frac{d_{*, *, e_{m,l}, 0}^{-(m,l)} + \varepsilon_{e_{m,l}}}{d_{*, *, *, 0}^{-(m,l)} + \sum_{i=1}^M \varepsilon_i} \quad (3.2)
\end{aligned}$$

$$\begin{aligned}
& p(x_{m,l}, y_{m,l} = 1 | y_{-(m,l)}, x_{-(m,l)}, w, z, e, \alpha, \beta, \gamma, \varepsilon, \rho) \\
\propto & (c_{x_{m,l}, m, *} + d_{x_{m,l}, m, *, *}^{-(m,l)} + \alpha_{x_{m,l}})(d_{*, m, *, 1}^{-(m,l)} + \rho_1) \times \frac{d_{x_{m,l}, *, e_{m,l}, 1}^{-(m,l)} + \gamma_{e_{m,l}}}{d_{x_{m,l}, *, *, 1}^{-(m,l)} + \sum_{i=1}^M \gamma_i} \quad (3.3)
\end{aligned}$$

In the above equations, $z_{m,n}$ denotes the topic of the n^{th} word for the m^{th} user, and $y_{m,l}$ is the reason indicator (content or non-content) of the l^{th} link for the m^{th} user. $w_{m,n}$, $x_{m,l}$ and $e_{m,l}$ follow similar definitions. Let $z_{-(m,n)}$ denote the topics for all words except $z_{m,n}$, and $y_{-(m,l)}$ and $x_{-(m,l)}$ follow an analogous definition. We define $c_{z,m,w}$ as the number of times word w is assigned to topic z for the m^{th} user, and $d_{x,m,e,y}$ as the number of times link e is assigned to topic x for the m^{th} user with indicator y . If any of the dimensions in above notations is not limited to a specific value, we use $*$ to denote. Essentially, $*$ represents an aggregation on the corresponding dimension. For example, $c_{z,*,w}$ is the total number of times

word w is assigned to topic z in the entire document collection. Finally, let $c_{z,m,w}^{-(m,n)}$ be the same meaning of $c_{z,m,w}$ only with the n^{th} word for the m^{th} user excluded. Similarly, $d_{x,m,e,y}^{-(m,l)}$ is defined in the same way as $d_{x,m,e,y}$ only without the count for the l^{th} link for the m^{th} user.

After the sampling algorithm has run for an appropriate number of iterations (until the chain has converged to a stationary distribution), the estimates for the parameters of θ , φ , μ , σ and π can be obtained via the following equations:

$$\theta_{x|m} = \frac{c_{x,m,*} + d_{x,m,*,*} + \alpha_x}{c_{*,m,*} + d_{*,m,*,*} + \sum_{i=1}^K \alpha_i} \quad (3.4)$$

$$\varphi_{w|z} = \frac{c_{z,*,w} + \beta_w}{c_{z,*,*} + \sum_{i=1}^W \beta_i} \quad (3.5)$$

$$\mu_{y|m} = \frac{d_{*,m,*,y} + \rho_y}{d_{*,m,*,*} + \rho_0 + \rho_1} \quad (3.6)$$

$$\sigma_{e|x} = \frac{d_{x,*,e,1} + \gamma_e}{d_{x,*,*,1} + \sum_{i=1}^M \gamma_i} \quad (3.7)$$

$$\pi_e = \frac{d_{*,*,e,0} + \varepsilon_e}{d_{*,*,*,0} + \sum_{i=1}^M \varepsilon_i} \quad (3.8)$$

3.4 Scalable Gibbs Sampling for FLDA

The rapid growth of microblog data poses a significant challenge for influence analysis in terms of both computation time and memory requirements. Scalable solutions that can take advantage of the computation power and memory capacity of multiple computers are becoming more crucial. However, the Gibbs sampling updates of FLDA shown in equations (3.1)-(3.3) are inherently sequential, which makes it very difficult to parallelize the computation. However, given the abundance of words and the large number of links in a microblog dataset, the dependency between different topic assignments or indicator assignments in equations (3.1)-(3.3) is relatively weak. As a result, we can relax the sequential requirement of the Gibbs sampling updates and distribute the computation to a number of processes running in parallel. In fact, similar observations were used

to develop approximate parallel/distributed Gibbs sampling algorithm for LDA in [82], [97] and [3]. We implemented our distributed FLDA Gibbs sampling algorithm on a distributed cluster computing framework called Spark [118]. Before the details of our distributed algorithm, we first provide a brief overview of Spark.

3.4.1 Spark Overview

Spark is a large-scale distributed processing framework specifically targeted at machine-learning iterative workloads. It uses a functional programming paradigm, and applies it on large clusters by providing a fault-tolerant implementation of distributed data sets called Resilient Distributed Data (RDD). RDDs can either reside in the aggregate main-memory of the cluster, or in efficiently serialized disk blocks. Especially for iterative processing, the opportunity to store the data in main-memory can significantly speed up processing. An RDD contains immutable data; i.e. it cannot be modified, however, a new RDD can be constructed by transforming an existing RDD.

The Spark runtime consists of a single coordinator node and multiple worker nodes. The coordinator keeps track of how to re-construct any partition of the RDD when any of the workers fails.

Computation in Spark is expressed using functional transformations over RDDs. For instance, assume that we have a log file, and that we want to transform each string to lower case. Consider the first two lines of actual Spark code in Listing 3.1: The first line of code defines an RDD of strings, called `baseRDD`, over a file “baseData.log” stored in a Hadoop Distributed FileSystem; each text line of the log file, corresponds to a string of the RDD. The second line of code, uses the `map` function to transform each string in `baseRDD` through the function `String.toLowerCase`. The transformation happens in parallel on all the workers, and defines a new RDD, called `lowerRDD` that contains the lower-case string of

```
1  val baseRDD = sc.textFile("hdfs://master/baseData.log")
2  val lowerRDD = lines.map(String.toLowerCase _)
3  val regexB = sc.broadcast(REGEX)
4  val nMatches = sc.accumulator(0)
5  lowerRDD.foreach (s =>
6    if ( s.matches(regexB.value) )
7      nMatches += 1
8  )
9  println("#Matches is:\%d".format(nMatches.value))
```

Listing 3.1: Sample Spark code.

each string in `baseRDD`.

Spark’s programming model provides additionally two useful abstractions: broadcast variables and accumulators. Broadcast variables are initialized at the coordinator node, and made available to all worker nodes, through efficient network broadcast algorithms. Spark chooses a topology-aware network-efficient algorithm to disseminate the data. Line 3 in Listing 3.1 initializes a broadcast variable called `regexB` to a regular expression (called `REGEX`). In Line 6, this value is used inside the `foreach` loop to check if any of the lines in the RDD called `lowerRDD` matches that regular expression. Note that broadcast variables are immutable, read-only, objects and cannot be modified by the workers.

Similar to a broadcast variable, an accumulator is also a variable that is initialized on the coordinator node, and sent to all the worker nodes. However, unlike a broadcast variable, an accumulator is mutable and can be used to aggregate results of computations at worker nodes. Worker nodes may update the state of the accumulator (usually just by incrementing it, or by using computations such as count and sum). At the end of the RDD transformation, each worker node sends its locally-updated accumulator back to the coordinator node, where all the accumulators are combined (using either a default or user-supplied combine

function) into a final result. In our example listing, `nMatches` is an accumulator that is locally incremented by all workers (line 7) before it is globally aggregated (through an implicit addition over all the partial results in line 9).

3.4.2 Distributed FLDA using Spark

We now describe how we use the Spark framework to implement the distributed Gibbs sampling algorithm for FLDA. In particular we discuss technical issues that distributed approaches encounter, and propose solutions and justify various implementation choices.

First, we define the notion of a *user object*. Each user object corresponds to a single user m , and holds information about the content (i.e. the actual words used by m) and the link structure (i.e. other users that m is following). For each word w and link e , the user object holds the last topic assignment, i.e. the corresponding latent variables z and x . For each link additionally it holds the last binary state (i.e. content-related or content-independent) for the y latent variable. Finally, each user object holds the *user-local* counters $d_{x,m,e,y}$, $c_{x,m,w}$, as well as all aggregates of these (like $d_{*,m,*,*}$) that show up in equations (1)-(3). Note that the corresponding local aggregates always have a m index in the subscript. Such aggregates are entirely local to a user and need not be shared.

Note that in addition to the user-local counters and aggregates, equations (1)-(3) require *global* aggregates (like $d_{x,*,*,1}$) over all the users. Such global aggregates always have a $*$ instead of m in the corresponding subscript index.

Based on previous work ([97]), the global aggregates are not stored in the user object, but are computed periodically and distributed to all workers through an accumulator. The idea is that such aggregates should change slowly and thus any inaccuracies (because of the periodic synchronization) shouldn't affect the quality of the final result. Although this assumption has been shown to work well

in practice for basic LDA, it is not evidently clear whether it works for FLDA. A big difference is that the global aggregates, that distributed LDA periodically needs to synchronize, are only per document terms and thus their count is limited (especially after typical pre-processing, like stop-word removal or stemming). However, distributing FLDA requires the periodic synchronization of orders of magnitude more aggregates; not only per document terms but also per user terms (and typically there are many more users). In the experiments, we show that for real datasets with millions of users, distributing FLDA still works very well.

Second, we define a mapping function, `GibbsSampleMap`, which takes as input one such user object, runs Gibbs sampling once and returns a new user object. In particular, this function goes over all the words and links in the object and (a) “undoes” the effects of the last assignment to the latent variables x , y and z (by properly decreasing the corresponding counts $d_{x,m,e,y}$, $c_{x,m,w}$ as well as all the corresponding local and global aggregates), (b) computes the new probabilities for the latent variables x , y and z according to the equations (1)-(3), and finally (c) assigns new latent variables according to these probabilities, and increases the corresponding counts and all user-local and global aggregates.

Putting all these together, first we initialize an RDD of user objects by (a) properly parsing and co-grouping the content and the link structure for each user, (b) randomly initializing the latent variable assignments and (c) computing the corresponding user-local counters and aggregates based on these initial assignments. Then we run a number of iterations over the RDD, where each iteration maps all user objects (in parallel) to new user objects using the `GibbsSampleMap` function we defined above. At the beginning of each iteration we accumulate and broadcast the global aggregates. We note, that each worker has its each own copy of the global aggregates, that the mapping function modifies. Thus although each worker starts with the same global aggregates, as user objects are transformed through the mapping functions, the workers’ copies of the global aggregates get

“out-of-sync”, until the start of the next iteration when new global aggregates are computed and broadcasted. Finally, when all the iterations are done, we use equations (4)-(8) to estimate the parameters θ , φ , μ , σ and π of the FLDA model.

3.4.3 Discussion

The distributed Gibbs sampling algorithm is quite generic and could be used to train other Bayesian models as well. We emphasize that the final result depends on the assumptions made in [82] and [97]. In particular, the global-aggregates change slowly and thus are not updated continuously, but only once every iteration (or even less often and asynchronously in the case of [97]). This choice does not seem to affect the final result of topic models like LDA. The literature shows that the quality of the result (in terms of perplexity or log-likelihood) is equivalent to that from a purely sequential implementation (where the global aggregates are always updated). We empirically show that the same holds for FLDA. We emphasize that it is not immediately clear whether this approach works for FLDA, since it requires the synchronization of order of magnitude more global aggregates. In Section 3.6 we compare the resulting user rankings produced by serial and distributed version of the algorithm and show virtually no difference. Finally we note that in the experiments with various real datasets, we observed that computing and synchronizing the global-aggregates just once every ten iterations doesn't seem to affect the quality of the results. It is an open problem exactly how infrequent such global updates can be.

Although Spark provides a lineage based fault-recovery mechanism, we chose to complement it using manual checkpoints for every ten iterations. The reason, is that replaying all the iterations from the beginning for every failed worker (although infrequent) takes quite some time. With the checkpoints, we guarantee that at most ten iterations will have to be replayed in case of failures. Our choice was also based on the fact that the cost of a checkpoint was negligible (a small

fraction of the time required to do an iteration).

Finally, we took extra care for the correctness of the distributed Monte-Carlo simulation. Since multiple workers are spawned at roughly the same time, typical random-number generators are seeded with similar (or even exactly the same) seeds. This introduces correlations between the pseudo random numbers generated across the workers. In extreme cases, two workers could “see” exactly the same stream of pseudo-random numbers. Such correlation jeopardizes the quality of the returned results. To guarantee correctness of the distributed simulation, we use the technique discussed in [53] for generating multiple streams of uniform numbers that are provably independent. In particular, we assign a unique stream for each RDD block and iteration pair (i.e. if we have 100 RDD blocks and 500 iterations, we have 100×500 independent streams of random numbers). This approach guarantees not only the correctness of the simulation, but also repeatability; every time we run the simulation with the same initial seed we get exactly the same results, regardless of the number of workers or possible worker failures.

3.5 Querying Topical Influencers

Finally, we propose a general search framework for topic-specific key influencers, called **SKIT**. Inspired by the popular search engine framework, SKIT allows a user to freely express his/her interests by typing a set of keywords. Then, SKIT returns an ordered list of key influencers by their influence scores that satisfy the user’s intent.

SKIT flexibly allows plugging in different topical influence methods. All that it needs from the underlying influence analysis are (a) the derivation of interested topics from the query keywords, and (b) the per-topic influence scores for every microblog user. More specifically, given a set of key words as a query q , SKIT first derives a weight $W(t, q)$ for each topic t in the set of all topics T , indicating the

likelihood of topic t being represented by query q . Then, utilizing the learned per-topic influence score for each user $\text{INFL}(t, u)$, the final influence score $\text{INFL}(q, u)$ for a user u given a query q is computed as

$$\text{INFL}(q, u) = \sum_{t \in T} W(t, q) \cdot \text{INFL}(t, u). \quad (3.9)$$

Finally, the users are returned in decreasing order of their influence scores $\text{INFL}(q, u)$.

When our FLDA model is used as the underlying topic-specific influence analysis method, the probability distributions $\theta_{z|m}$ and $\sigma_{e|x}$ are produced as part of the results. Here, $\theta_{z|m}$ represents the probability of topic z given user m , and $\sigma_{e|x}$ is the probability of user e being followed by someone given topic x . If we treat a query q as a new user, we can use the folding-in [49] or the variational inference [97] technique on FLDA to quickly learn $\theta_{z=t|m=q}$, the probability of topic t given the query q , and use this value as $W(t, q)$ in Equation (3.9). On the other hand, the per-topic influence score $\text{INFL}(t, u)$ for each user can be quantified by $\sigma_{e=u|x=t}$.

Besides FLDA, our flexible SKIT search framework can also easily plug in Link-LDA, TSPR and TwitterRank. The folding-in and the variational inference techniques equally apply to Link-LDA and LDA, if LDA is used in the topic modeling preprocess for TSPR and TwitterRank to compute $W(t, q)$. The definition of $\text{INFL}(t, u)$ for Link-LDA is the same as in FLDA. For both TSPR and TwitterRank, $\text{INFL}(t, u)$ is simply the PageRank score for user u and topic t .

3.6 Experiments

In this section, we start with evaluating the effectiveness of our FLDA model on two microblog datasets, Twitter and Tencent Weibo. On the Twitter dataset, we give examples of topics and influencers found by the FLDA model, then we use the Tencent Weibo dataset to systematically compare FLDA with a number

Table 3.2: Statistics of Experimental Datasets.

Dataset	# users	# dist. words	# total words	# links
Twitter	1.76 M	159 K	2363 M	183 M
Weibo	2.33 M	714 K	492 M	51 M

Table 3.3: A sample of FLDA topics and their influencers.

Topic	Top-10 keywords	Top-5 influencers
“Information Technology”	data, web, cloud, software, open, windows, microsoft, server, security, code	Tim O’Reilly, Gartner Inc., Scott Hanselman (software blogger), Jeff Atwood (software blogger, co-founder of stackoverflow.com), Elijah Manor (software blogger)
“Food and drink”	food, chocolate, coffee, eat, chicken, lunch dinner, cheese, recipe, tea	Whole Foods (organic grocery chain), Foodimentary.com (food blog), WineTwits.com (wine community), Barack Obama, L.A. Times Food
“Cycling and running”	bike, ride, race, training, running, miles, team, workout, marathon, fitness	Lance Armstrong, Levi Leipheimer, George Hincapie (all 3, US Postal pro cycling team members), Johan Bruyneel (US Postal team director), RSLT (radioshackleopardtrek.com – pro cycling team)
“Advertiser’s dream”	class, sleep, hate, bed, tired, movie, homework, finally, bored, ugh	Taylor Swift, Pete Wentz, Katy Perry (all 3 singers), Perez Hilton (celeb blogger), Lady Gaga
“Ppl can’t spell”	ppl, dnt, nite, tht, jus, goin, lov, wat, abt, plz	Kim Kardashian, Kourtney Kardashian, Khloe Kardashian, Tila Tequila (all 4 reality TV stars), Ciara (singer)
“Down under”	travel, Australia, latest, Sydney, Melbourne fishing, Australian, trip, hotel, island	Kevin Rudd (Australian PM in 2010), Rove McManus, Dave Hughes, Wil Anderson (all three are Aussie comedians and TV hosts), Ruby Rose (Australian model and TV presenter)

of existing approaches including TSPR, TwitterRank and Link-LDA. Finally, we demonstrate the scalability of the distributed Gibbs sampling algorithm, and show that it produces results with indistinguishable quality as the sequential algorithm.

Experiment Setup. The sequential FLDA Gibbs sampling algorithm was run on an 4-core Intel Xeon (X5672) 64-bit 3.2GHz server with 192GB RAM. For distributed FLDA Gibbs sampling, we used a cluster of 27 IBM System x iDataPlex dx340 servers. Each server consisted of two quad-core Intel Xeon (E5540) 64-bit 2.5GHz processors, 32GB RAM, and interconnected using 1GB Ethernet. We reserved one server as the Spark coordinator, and use the remaining ones for workers. Each machine was configured to run up to 8 concurrent workers. By default, we used 200 workers for distributed FLDA.

Table 3.4: A sample of the topics of FLDA and Link-LDA with their influencers

Topic	Model	Top-10 keywords	Top-5 influencers
"Job"	FLDA	business, job, jobs, management, manager, sales, services, company, service, hiring	job-hunt.org, jobsguy.com, integritystaffing.com/blog (Job Search Ninja), JobConcierge.com, careerealism.com
	Link-LDA	job, jobs, manager, business, sales, management, service, company, services, hiring	Paul Terry Walhus (web host developer and blogger), Wayne Sutton (startup advisor), Adam Glickman (tech professional), bloggersblog.com (blogging news), Mary Hodder (tech blogger)
"Music"	FLDA	listening, album, rock, band, song, john, black, top, tour, artists	pitchfork.com (music website), Trent Reznor (singer), Paste Magazine (music magazine), New Musical Express, Sub Pop Records
	Link-LDA	listening, rock, album, song, band, black, john, top, beatles, bob	Club Ubuntu, Nithin Jawali (tech enthusiast), Paul Shaffer (musician) Debra Zimmer (consultant), iheartquotes.com (a collection of quotes)
"Justice"	FLDA	police, court, case, law, report, death, story, arrested, woman, state	Barack Obama, CNN Breaking News, The New York Times, CanadaCool.com, BBC Breaking News
	Link-LDA	police, ap, law, court, report, press, case, death, reuters, state	Health Brand, 2humor.com (funny stuff), healthsmartme.tumblr.com Daniel Vega (lawyer), Multiplaza Shopping (shopping deals)
"Weather"	FLDA	snow, weather, rain, winter, high, nc, denver, storm, wind, county	CNN Breaking News, The Denver Post, NPR News, The Weather Channel, CBS Denver
	Link-LDA	snow, weather, atlanta, rain, high, nc, county, south, fire, north	DiningPerks.com, Georgia Aquarium, Atlanta Journal-Constitution (Atlanta newspaper), HelloNorthGeorgia.com, Q100 Atlanta (radio)

3.6.1 Effectiveness on Twitter Dataset

We first evaluate our FLDA model on a Twitter dataset¹, crawled between October 2009 and January 2010. The raw dataset consists of roughly half a terabyte of text and link information. The basic statistics of this dataset are given in Table 7.2. We used the tokenizer from the TweetNLP project [46] in order to improve the accuracy of the recognized terms in the noisy text. We tried to further reduce the inherent noise of tweets, by removing terms that appear in less than 50 tweets. We set the number of topics to 100 and run the distributed FLDA Gibbs sampling for 500 iterations. All the priors were set to 0.1 except ρ which was set to 1. These settings are fairly typical for LDA-based approaches and their tuning is beyond the scope of this work.

Table 3.3 shows some of the resulting topics with their top keywords and influencers. We named these topics to simplify the presentation. Intuitively, it is clear that the influencers are very relevant to the corresponding topics. For example, one would expect O’Reilly publishers, Gartner research, and popular

¹This dataset was crawled in a BSF manner with the top 1000 users in `twitterholic.com` as the seeds.

software bloggers to be influential for an IT-related topic. Just as one would expect school age kids to be influenced by pop stars. Some of the findings are insightful. For example, Australians seem to be particularly influenced by comedians. While the first person in the list is Prime Minister of Australia (at the time of the crawl), the following three are comics.

FLDA separated the “globally” popular users from the content-specific influencers, and elected that 15% of all links were content-independent. In other words, 15% of the time, these popular users were followed regardless of what people tweet about. By comparison, the largest topic was associated with less than 2.5% of all links. The top five globally popular users detected by FLDA were: Pete Wentz (singer), Ashton Kutcher (actor), Greg Grunberg (actor, author of Yowza mobile app), Britney Spears (singer), and Ellen DeGeneres (comedian, TV host). In comparison, the top five most-followed Twitter accounts were: Barack Obama, Ashton Kutcher, Britney Spears, Ellen DeGeneres, Shaquille O’Neal (basketball player). Although President Obama was most followed, we found his impact was most prominent in politics, which was topic-related. Similarly the basketball star Shaquille O’Neal was mostly followed due to his impact in basketball. FLDA can correctly identify topic-specific influence from the content-independent popularity.

There were a number of similar topics produced by both FLDA and Link-LDA. Table 3.4 compares the top five influencers from FLDA and Link-LDA for a few example topics. As shown from the table, FLDA produced dramatically better results than Link-LDA. For example, the “Jobs” topic produced by FLDA and Link-LDA had virtually the same top-10 keywords. The top five influencers identified by FLDA were all popular job-search websites, whereas the influencers found by Link-LDA were mostly tech bloggers. Upon inspection of their tweets it seems clear that the Link-LDA list was much less relevant to the job search topic. As another example, on the “Music” topic, FLDA successfully identified popular music media as influencers, whereas Link-LDA misidentified Club Ubuntu and a

consultant as influencers in the music topic.

Naturally, such anecdotal evidence is very hard to generalize and quantify. Luckily, 2012 KDD Cup provided us with the data needed to objectively measure the quality of FLDA and other approaches, as we describe next.

3.6.2 Effectiveness on Tencent Weibo Dataset

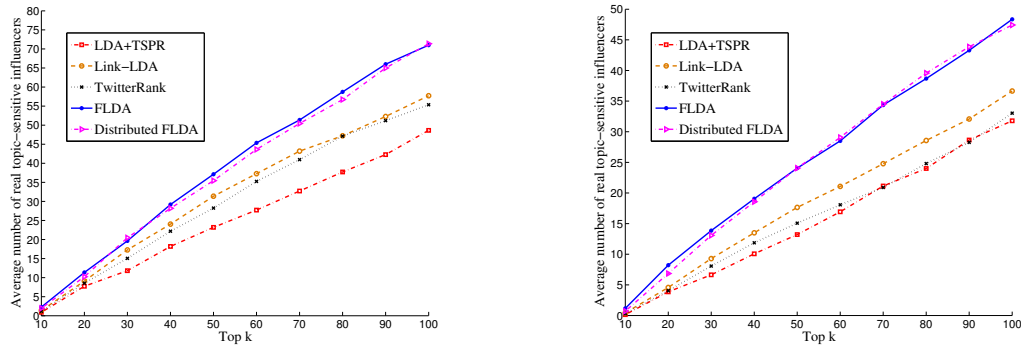
In this section, we systematically evaluate the effectiveness of our FLDA model on a sample dataset from the popular Chinese microblog site – Tencent Weibo (t.qq.com).

This Tencent Weibo dataset is released by KDD Cup 2012². The basic statistics of this dataset ³ are given in Table 7.2. A very nice feature of the Weibo dataset is the set of provided VIP users (also called items in Weibo), which enables us to systematically evaluate the precision of various key influencer methods. These VIP users are manually labeled by Weibo administrators, and organized in hierarchical categories. An example hierarchical category is *science_and_technology.internet.mobile*, where categories in different levels are separated by a dot “.”. In this dataset, categories are anonymized as integers, such as 1.4.2.3. There are 377 categories and on average each category contains 16.2 VIPs. According to Weibo, the VIP users are typically famous people and organizations. In other words, they are “key influencers” in their corresponding categories. As a result, the VIP users can be used as the “ground truth” for our empirical evaluation. While we don’t expect VIP categories to have 100% precision or recall, they give us enough information to facilitate relative comparisons across different schemes.

Based on this information, we have set up the following experiment. For a given category, we use one VIP (i.e. all the words of this user) as the query, and

²www.kddcup2012.org/c/kddcup2012-track1/data

³In the Tencent Weibo dataset, for each user, the appearance of each word is associated with a weight (usually ≤ 1.0). We multiply this weight by 100 to approximate the underlying word frequency.



(a) Category 1.6.2.1 (#VIPs in this category: 277) (b) Category 1.6.2.2 (#VIPs in this category: 123)

Figure 3.3: Average number of returned VIPs

observe how many of the fellow VIP users in the same category are identified as top influencers by the different schemes. In the following, we compare our FLDA model with TSPR, TwitterRank and Link-LDA. We maintain the number of topics at 100 for all the methods and run 500 iterations for LDA (used in TSPR and TwitterRank), Link-LDA and FLDA. The priors used are 1.0 for α , 0.01 for β , γ and ϵ , and 0.1 for ρ .

Figure 3.3(a) and Figure 3.3(b) compare our FLDA model with TSPR, TwitterRank and Link-LDA on two of the largest categories in the Weibo dataset. For each category, we use every VIP user as a key influencer query and check how many of the top K returned users are the fellow VIP users. We report the average number of VIPs among the top K returned results across all the queries. As shown in both figures, our FLDA model consistently produces better precision results than the others by a significant margin. TSPR is usually the worst among all methods, followed by TwitterRank. Link-LDA performs slightly better than the two PageRank-based approaches.

To analyze the results across all categories we employ a standard Mean Average Precision (MAP) [76] metric. MAP for a set of queries is defined as the mean of

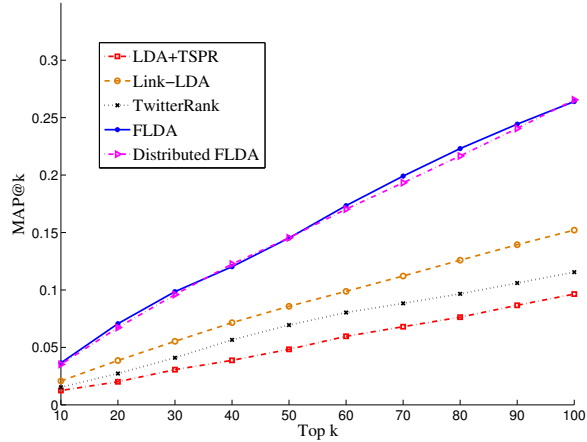


Figure 3.4: Mean Average Precision

the average precision scores ($AvgP$) for each query. $AvgP$ of a list of top- k query results is defined as the average of precision values for all k prefixes.

Figure 3.4 shows MAP of all the queries across all categories in the Weibo dataset. Again, FLDA produces significantly better results than the competing methods, more than 2 times better than TSPR and TwitterRank, and around 1.6 times better than Link-LDA. Interestingly, FLDA elected only 50% of Weibo links to be content-related. This explains the significant advantage of FLDA over Link-LDA, which assumes that all links are topic-specific.

As shown in Figure 3.3(a), 3.3(b) and Figure 3.4, the distributed FLDA consistently produces result with quality almost identical to that of the sequential FLDA. This confirms the relaxed dependency assumption on which our distributed FLDA Gibbs sampling is based.

Throughout the experiments we measured the time taken by the on-line component of our SKIT search framework. On average, each query takes 1.7 sec to get the results, and this time does not depend on the off-line modeling scheme we use.

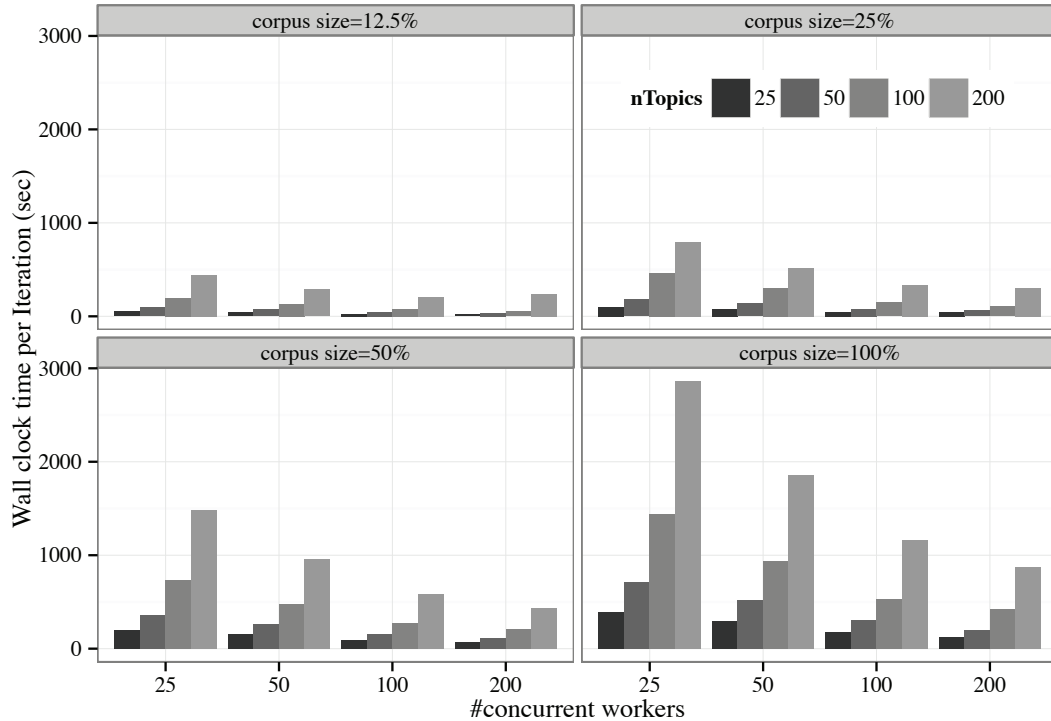


Figure 3.5: Speed-Up of Distributed FLDA on the Twitter dataset.

3.6.3 Scalability

Before evaluating the scalability of our distributed FLDA Gibbs sampling algorithm, we first report the execution times for the sequential algorithm. For the KDD Weibo dataset, the sequential Gibbs sampling on a high-end server (192GB RAM, 3.2GHz processor) takes around 13 minutes per iteration, and for the Twitter dataset, it takes more than one hour per iteration. On Twitter dataset FLDA runs longer because there are many more words and links to sample. Running sequential Gibbs sampling for 500 iterations takes around 4.6 days for the KDD Weibo dataset, and would take 21 days for the Twitter dataset! This clearly motivates the need for a scalable solution.

Our distributed algorithm completes 500 iterations on Twitter data in about 36 hours, using 200 workers on 27 machines. Overall, the distributed FLDA in this instance is about 14 times faster than a sequential implementation running

on single, large-memory server.

We tested the scalability of the distributed algorithm along three dimensions: data size, number of topics, and the number of concurrent workers. To obtain the scaled down dataset we performed uniform random sampling of users, for example to generate a 4 times smaller dataset we use a sampling rate of 25%. The results are summarized in Figure 3.5 where the scaled-down dataset is denoted as *corpus size* and is measured by the sampling rate used. We explore a wide range of sizes (from 12.5% all the way up to 100%), number of topics (from 25 to 200) and number of workers (from 25 to 200). The figure shows that the distributed FLDA scales well along all dimensions, given the limitations of our cluster. Our (older) CPU's were significantly oversubscribed with 8 workers per node, which was the case with 200 workers.

3.7 Conclusion

This work addresses the problem of identifying topic-specific key influencers in microblog networks. To model the per-topic influence of each user, we introduce a novel Bernoulli-Multinomial mixture model called FLDA. FLDA incorporates the content of tweets and the network structure of microblogs into one unified model. Different from the previous work, such as Link-LDA, our FLDA model is specifically designed for microblogs in that it captures the fact that in reality a user sometimes follows another due to content-independent reasons. Moreover, in order to apply FLDA to a web-scale microblog network, we design a distributed Gibbs sampling algorithm for FLDA on the Spark distributed computing framework. Finally, the FLDA model is incorporated in a proposed general search framework for topic-specific key influencers, which provides a keyword search interface for users to freely query key influencers in different topic combinations.

Through experiments on two real-world microblog datasets, we demonstrate

that FLDA significantly outperforms the state-of-the-art methods in terms of precision. Furthermore, the distributed Gibbs sampling algorithm for FLDA provides excellent speed-up to hundreds of workers.

Appendix

The Gibbs sampling equation for latent variable z can be derived in a similar way to Link-LDA. We omit its derivation due to space limitation. Let us derive the posterior probability of latent variables x and y :

$$\begin{aligned}
& p(x_{m,l}, y_{m,l} | y_{-(m,l)}, x_{-(m,l)}, w, z, e, \alpha, \beta, \gamma, \varepsilon, \rho) \\
& \propto p(x, y, w, z, e | \alpha, \beta, \gamma, \varepsilon, \rho) \\
& = \int \int \int \int \int p(x, y, w, z, e, \theta, \varphi, \sigma, \pi, \mu | \alpha, \beta, \gamma, \varepsilon, \rho) d\theta d\varphi d\sigma d\pi d\mu \\
& = \int \int \int \int \int p(x|\theta)p(y|\mu)p(w|z, \varphi)p(z|\theta)p(e|x, y, \sigma, \pi) \\
& \quad \times p(\theta|\alpha)p(\varphi|\beta)p(\sigma|\gamma)p(\pi|\varepsilon)p(\mu|\rho) d\theta d\varphi d\sigma d\pi d\mu
\end{aligned} \tag{3.10}$$

As $p(e|x, y, \sigma, \pi) = p(e|x, \sigma)^y p(e|\pi)^{1-y}$, we get,

$$\begin{aligned}
& = \int p(\theta|\alpha)p(z|\theta)p(x|\theta)d\theta \int p(\sigma|\gamma)p(e|x, \sigma)^y d\sigma \\
& \quad \times \int p(\varphi|\beta)p(w|z, \varphi)d\varphi \int p(\pi|\varepsilon)p(e|\pi)^{1-y} d\pi \int p(\mu|\rho)p(y|\mu)d\mu
\end{aligned} \tag{3.11}$$

Let us derive the first two integrals in Equation (3.11).

$$\begin{aligned}
& \int p(\theta|\alpha)p(z|\theta)p(x|\theta)d\theta \int p(\sigma|\gamma)p(e|x, \sigma)^y d\sigma \\
& = \int \prod_{j=1}^M p(\theta_j|\alpha) \prod_{j=1}^M \prod_{u=1}^{N_j} p(z_{j,u}|\theta_j) \prod_{j=1}^M \prod_{v=1}^{L_j} p(x_{j,v}|\theta_j) d\theta \\
& \quad \times \int \prod_{k=1}^K p(\sigma_k|\gamma) \prod_{j=1}^M \prod_{v=1}^{L_j} p(e_{j,v}|\sigma_{x_{j,v}})^{y_{j,v}} d\sigma
\end{aligned} \tag{3.12}$$

Expand each probability formula based on its density,

$$\begin{aligned}
&= \int \prod_{j=1}^M \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K \theta_{j,k}^{\alpha_k-1} \prod_{j=1}^M \prod_{u=1}^{N_j} \theta_{z_j,u} \prod_{j=1}^M \prod_{v=1}^{L_j} \theta_{x_j,v} d\theta \\
&\quad \times \prod_{k=1}^K \int \frac{\Gamma(\sum_{i=1}^M \gamma_i)}{\prod_{i=1}^M \Gamma(\gamma_i)} \prod_{i=1}^M \sigma_{k,i}^{\gamma_i-1} \prod_{j=1}^M \prod_{v=1}^{L_j} \sigma_{x_j,v,e_{j,v}}^{y_{j,v}} d\sigma_k
\end{aligned} \tag{3.13}$$

Replace the innermost products over words in a document N_m by exponentiating to the sum of the counts, and do the same replacement for the products over users,

$$\begin{aligned}
&= \prod_{j=1}^M \int \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K \theta_{j,k}^{\alpha_k-1} \prod_{k=1}^K \theta_{j,k}^{c_{k,j,*}} \prod_{k=1}^K \theta_{j,k}^{d_{k,j,*,*}} d\theta_j \\
&\quad \times \prod_{k=1}^K \int \frac{\Gamma(\sum_{i=1}^M \gamma_i)}{\prod_{i=1}^M \Gamma(\gamma_i)} \prod_{i=1}^M \sigma_{k,i}^{\gamma_i-1} \prod_{i=1}^M \sigma_{k,i}^{d_{k,*,i,1}} d\sigma_k \\
&= \prod_{j=1}^M \int \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K \theta_{j,k}^{\alpha_k + c_{k,j,*} + d_{k,j,*,*} - 1} d\theta_j \\
&\quad \times \prod_{k=1}^K \int \frac{\Gamma(\sum_{i=1}^M \gamma_i)}{\prod_{i=1}^M \Gamma(\gamma_i)} \prod_{i=1}^M \sigma_{k,i}^{\gamma_i + d_{k,*,i,1} - 1} d\sigma_k
\end{aligned} \tag{3.14}$$

Multiply each term by a constant equal to one (consisting of two inverse fractions), and distribute the integral over the original constant Γ -function fraction for the priors,

$$\begin{aligned}
&= \prod_{j=1}^M \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \frac{\prod_{k=1}^K \Gamma(\alpha_k + c_{k,j,*} + d_{k,j,*,*})}{\Gamma(\sum_{k=1}^K \alpha_k + c_{k,j,*} + d_{k,j,*,*})} \\
&\quad \times \int \frac{\Gamma(\sum_{k=1}^K \alpha_k + c_{k,j,*} + d_{k,j,*,*})}{\prod_{k=1}^K \Gamma(\alpha_k + c_{k,j,*} + d_{k,j,*,*})} \prod_{k=1}^K \theta_{j,k}^{\alpha_k + c_{k,j,*} + d_{k,j,*,*} - 1} d\theta_j \\
&\quad \times \prod_{k=1}^K \frac{\Gamma(\sum_{i=1}^M \gamma_i)}{\prod_{i=1}^M \Gamma(\gamma_i)} \frac{\prod_{i=1}^M \Gamma(\gamma_i + d_{k,*,i,1})}{\Gamma(\sum_{i=1}^M \gamma_i + d_{k,*,i,1})} \\
&\quad \times \int \frac{\Gamma(\sum_{i=1}^M \gamma_i + d_{k,*,i,1})}{\prod_{i=1}^M \Gamma(\gamma_i + d_{k,*,i,1})} \prod_{i=1}^M \sigma_{k,i}^{\gamma_i + d_{k,*,i,1} - 1} d\sigma_k
\end{aligned} \tag{3.15}$$

Note that both integrals are over the entire support of Dirichlet densities, so they both evaluate to 1, and hence drop out of the products,

$$\begin{aligned}
&= \prod_{j=1}^M \frac{\Gamma(\sum_{k=1}^K \alpha_k) \prod_{k=1}^K \Gamma(\alpha_k + c_{k,j,*} + d_{k,j,*,*})}{\prod_{k=1}^K \Gamma(\alpha_k) \Gamma(\sum_{k=1}^K \alpha_k + c_{k,j,*} + d_{k,j,*,*})} \\
&\quad \times \prod_{k=1}^K \frac{\Gamma(\sum_{i=1}^M \gamma_i) \prod_{i=1}^M \Gamma(\gamma_i + d_{k,*,i,1})}{\prod_{i=1}^M \Gamma(\gamma_i) \Gamma(\sum_{i=1}^M \gamma_i + d_{k,*,i,1})} \tag{3.16}
\end{aligned}$$

Eliminate constant terms that do not depend on the position (m, l) ,

$$\propto \frac{\prod_{k=1}^K \Gamma(\alpha_k + c_{k,m,*} + d_{k,m,*,*})}{\Gamma(\sum_{k=1}^K \alpha_k + c_{k,m,*} + d_{k,m,*,*})} \times \prod_{k=1}^K \frac{\Gamma(\gamma_{e_{m,l}} + d_{k,*,e_{m,l},1})}{\Gamma(\sum_{i=1}^M \gamma_i + d_{k,*,i,1})} \tag{3.17}$$

Define $c^{-(m,l)}$ the same way as c , only without the counts for position (m, l) . Then, for counts that do not include position (m, l) , $c^{-(m,l)} = c$. For ones that do include (m, l) , $c^{-(m,l)} = c + 1$. $d^{-(m,l)}$ is defined in the same way. Also, using the fact that $\Gamma(x + 1) = x \times \Gamma(x)$, expand out the incremented terms depending on (m, l) ,

$$\begin{aligned}
&= \frac{\prod_{k \neq x_{m,l}} \Gamma(\alpha_k + c_{k,m,*} + d_{k,m,*,*}^{-(m,l)})}{\Gamma(1 + \sum_{k=1}^K \alpha_k + c_{k,m,*} + d_{k,m,*,*}^{-(m,l)})} \\
&\quad \times \Gamma(\alpha_{x_{m,l}} + c_{x_{m,l},m,*} + d_{x_{m,l},m,*,*}^{-(m,l)}) \\
&\quad \times (\alpha_{x_{m,l}} + c_{x_{m,l},m,*} + d_{x_{m,l},m,*,*}^{-(m,l)}) \\
&\quad \times \prod_{k \neq x_{m,l}} \frac{\Gamma(\gamma_{e_{m,l}} + d_{k,*,e_{m,l},1}^{-(m,l)})}{\Gamma(\sum_{i=1}^M \gamma_i + d_{k,*,i,1})} \times \frac{\Gamma(\gamma_{e_{m,l}} + d_{x_{m,l},*,e_{m,l},1}^{-(m,l)})}{\Gamma(\sum_{i=1}^M \gamma_i + d_{x_{m,l},*,i,1}^{-(m,l)})} \\
&\quad \times \frac{\gamma_{e_{m,l}} + d_{x_{m,l},*,e_{m,l},1}^{-(m,l)}}{\sum_{i=1}^M (\gamma_i + d_{x_{m,l},*,i,1}^{-(m,l)})} \tag{3.18}
\end{aligned}$$

Refold the residual Γ -function terms back into their general products,

$$\begin{aligned}
&= \frac{\prod_{k=1}^K \Gamma(\alpha_k + c_{k,m,*} + d_{k,m,*,*}^{-(m,l)})}{\Gamma(1 + \sum_{k=1}^K \alpha_k + c_{k,m,*} + d_{k,m,*,*}^{-(m,l)})} \\
&\quad \times (\alpha_{x_{m,l}} + c_{x_{m,l},a,*} + d_{x_{m,l},m,*,*}^{-(m,l)}) \\
&\quad \times \prod_{k=1}^K \frac{\Gamma(\gamma_{e_{m,l}} + d_{k,*,e_{m,l},1}^{-(m,l)})}{\Gamma(\sum_{i=1}^M \gamma_i + d_{k,*,i,1})} \times \frac{\gamma_{e_{m,l}} + d_{x_{m,l},*,e_{m,l},1}^{-(m,l)}}{\sum_{i=1}^M (\gamma_i + d_{x_{m,l},*,i,1}^{-(m,l)})} \tag{3.19}
\end{aligned}$$

Remove all the terms that do not depend on $x_{m,l}$ or $y_{m,l}$.

If $y_{m,l} = 0$, the last term $\frac{\gamma_{e_{m,l}} + d_{x_{m,l},*,e_{m,l},1}^{-(m,l)}}{\sum_{i=1}^M (\gamma_i + d_{x_{m,l},*,i,1}^{-(m,l)})}$ in Equation (3.19) does not exist,

$$\propto \alpha_{x_{m,l}} + c_{x_{m,l},m,*} + d_{x_{m,l},m,*,*}^{-(m,l)} \quad (3.20)$$

If $y_{m,l} = 1$,

$$\propto \frac{(\alpha_{x_{m,l}} + c_{x_{m,l},m,*} + d_{x_{m,l},m,*,*}^{-(m,l)}) (\gamma_{e_{m,l}} + d_{x_{m,l},*,e_{m,l},1}^{-(m,l)})}{\sum_{i=1}^M (\gamma_i + d_{x_{m,l},*,i,1}^{-(m,l)})} \quad (3.21)$$

The third integral $\int p(\varphi|\beta)p(w|z, \varphi)d\varphi$ in Equation (3.11) is independent of both x and y , so it can be safely canceled out. Let us turn to the fourth integral in Equation (3.11).

$$\begin{aligned} & \int p(\pi|\varepsilon)p(e|\pi)^{1-y}d\pi \\ &= \int \frac{\Gamma(\sum_{i=1}^M \varepsilon_i)}{\prod_{i=1}^M \Gamma(\varepsilon_i)} \prod_{i=1}^M \pi_i^{\varepsilon_i-1} \prod_{j=1}^M \prod_{v=1}^{L_j} \pi_{e_{j,v}}^{1-y_{j,v}} d\pi \\ &= \int \frac{\Gamma(\sum_{i=1}^M \varepsilon_i)}{\prod_{i=1}^M \Gamma(\varepsilon_i)} \prod_{i=1}^M \pi_i^{d_{*,*,i,0} + \varepsilon_i - 1} d\pi \\ &\propto \frac{\Gamma(\sum_{i=1}^M \varepsilon_i)}{\prod_{i=1}^M \Gamma(\varepsilon_i)} \times \frac{\prod_{i=1}^M \Gamma(d_{*,*,i,0} + \varepsilon_i)}{\Gamma(\sum_{i=1}^M d_{*,*,i,0} + \varepsilon_i)} \\ &\propto \frac{\prod_{i \neq e_{m,l}} \Gamma(d_{*,*,i,0} + \varepsilon_i) \times \Gamma(d_{*,*,e_{m,l},0} + \varepsilon_{e_{m,l}})}{\Gamma(\sum_{i=1}^M d_{*,*,i,0} + \varepsilon_i)} \quad (3.22) \end{aligned}$$

If $y_{m,l} = 0$, Equation (3.22) can be written as:

$$\begin{aligned} &= \frac{\prod_{i \neq e_{m,l}} \Gamma(d_{*,*,i,0}^{-(m,l)} + \varepsilon_i) \times \Gamma(d_{*,*,e_{m,l},0}^{-(m,l)} + \varepsilon_{e_{m,l}} + 1)}{\Gamma(1 + \sum_{i=1}^M d_{*,*,i,0}^{-(m,l)} + \varepsilon_i)} \\ &= \frac{\prod_i \Gamma(d_{*,*,i,0}^{-(m,l)} + \varepsilon_i)}{\Gamma(\sum_{i=1}^M d_{*,*,i,0}^{-(m,l)} + \varepsilon_i)} \times \frac{d_{*,*,e_{m,l},0}^{-(m,l)} + \varepsilon_{e_{m,l}}}{\sum_{i=1}^M d_{*,*,i,0}^{-(m,l)} + \varepsilon_i} \\ &\propto \frac{d_{*,*,e_{m,l},0}^{-(m,l)} + \varepsilon_{e_{m,l}}}{\sum_{i=1}^M d_{*,*,i,0}^{-(m,l)} + \varepsilon_i} \quad (3.23) \end{aligned}$$

If $y_{m,l} = 1$, Equation (3.22) can be written as:

$$= \frac{\prod_i \Gamma(d_{*,*,i,0}^{-(m,l)} + \varepsilon_i)}{\Gamma(\sum_{i=1}^M d_{*,*,i,0}^{-(m,l)} + \varepsilon_i)} \propto 1 \quad (3.24)$$

Finally, we derive the last integral in Equation (3.11).

$$\begin{aligned}
& \int p(\mu|\rho)p(y|\mu)d\mu \\
&= \prod_{j=1}^M \int \frac{\Gamma(\sum_s \rho_s)}{\prod_s \Gamma(\rho_s)} \prod_s \mu_{j,s}^{\rho_s-1} \prod_{j=1}^M \prod_{v=1}^{L_j} \mu_{j,y_{j,v}} d\mu_j \\
&= \prod_{j=1}^M \int \frac{\Gamma(\sum_s \rho_s)}{\prod_s \Gamma(\rho_s)} \prod_s \mu_{j,s}^{d_{*,j,*,s}+\rho_s-1} d\mu_j \\
&\propto \prod_{j=1}^M \frac{\prod_s \Gamma(d_{*,j,*,s} + \rho_s)}{\Gamma(\sum_s d_{*,j,*,s} + \rho_s)} \\
&= \prod_{j \neq m} \frac{\prod_s \Gamma(d_{*,j,*,s} + \rho_s)}{\Gamma(\sum_s d_{*,j,*,s} + \rho_s)} \times \frac{\prod_s \Gamma(d_{*,m,*,s} + \rho_s)}{\Gamma(\sum_s d_{*,m,*,s} + \rho_s)} \\
&\propto \frac{\prod_{s \neq y_{m,l}} \Gamma(d_{*,m,*,s}^- + \rho_s) \times \Gamma(d_{*,m,*,y_{m,l}}^- + \rho_{y_{m,l}} + 1)}{\Gamma(1 + \sum_s d_{*,m,*,s}^-)} \\
&= \frac{\prod_s \Gamma(d_{*,m,*,s}^- + \rho_s)}{\Gamma(\sum_s d_{*,m,*,s}^- + \rho_s)} \times \frac{d_{*,m,*,y_{m,l}}^- + \rho_{y_{m,l}}}{d_{*,m,*,*}^- + \sum_s \rho_s} \\
&\propto d_{*,m,*,y_{m,l}}^- + \rho_{y_{m,l}} \tag{3.25}
\end{aligned}$$

Finally, substituting Equations (3.20), (3.23) and (3.25) into Equation (3.11) gives Equation (3.2). Similary, substituting Equations (3.21), (3.24) and (3.25) into Equation (3.11) gives Equation (3.3).

CHAPTER 4

Bayesian Nonparametric Modeling for Microblog Data Analysis

4.1 Introduction

We have discussed above the methodology to analyze microblog data by Bayesian parametric modeling, where the number of topics has to be specified a priori. In this section, we present a different modeling paradigm, which is Bayesian nonparametric modeling. It allows the representation of the microblog data to grow structurally as more data are observed.

As opposed to the previous parametric model FLDA, the two nonparametric models described below are capable of letting the data speak for itself to automatically determine the number of topics needed in the models. Both models are able to integrate the analysis of tweet content and that of retweet behavior of users in the same probabilistic framework. Moreover, they both jointly model users' interest in tweet and retweet.

4.2 User-Retweet Model (URM)

Identifying users' interest in tweet and retweet is key for building user profiles to predict the user behavior and preference in various applications on Twitter, such as tweet recommendation and followee recommendation. Therefore, a Bayesian model, which properly captures the great diversity of user interests on Twitter, is

clearly needed. We refer to the first model as User-Retweet Model (URM).

Twitter has become a central nexus for discussion of the topics of the day. On Twitter, users from all over the world tweet a variety of topics of interest. Naturally, each user has distinct preference and topical interest. To characterize the heterogeneity among all users, we model each user as a unique mixture of a set of topics, where the mixing proportion governs his or her personal interest. In detail, each user possesses a distinct probability distribution over the topics, indicating the probability that he or she is interested in tweeting each individual topic. For example, consider a mini set of two topics: politics and food. One user may tweet the politics topic with a higher probability than the food topic, while another may be more interested in tweeting food than tweeting politics. Given a set of topics, a Twitter user generates each word in their tweets from one of the topics based on the distribution specific to this topic.

In addition to tweets, retweets convey useful clues about the users' interest and preference. If multiple users retweet a certain message, they are likely to have common topical interest reflected by this message. In order to capture the diversity of topics exhibited by retweets, we further model each retweet as a mixture of a set of topics. Specifically, each retweet is represented as a distribution over the topics, quantifying the probability of covering each individual topic.

In a mixture model, the number of mixture components is usually manually specified and empirically tuned to determine the granularity of the model. However, given the dynamic nature and large scale of retweet data, it is infeasible to manually exhaust the optimal number of topics in a retweet model. To address this limitation, we resort to a fully data-driven approach, i.e., imposing Dirichlet process priors over the mixture components [42], which allows the number of topics to be automatically determined based on the data characteristics.

4.2.1 Generative Process for URM

The problem of retweet modeling is to specify a probabilistic process by which the observed data, i.e., all the words in tweets, denoted by \mathbf{w} , and all the words in retweets, denoted by \mathbf{x} , may have been generated. In URM, we assume that in tweeting, to choose a word a user would first select a topic of interest according to his or her unique topic distribution, from which he or she would then pick a word w based on its generative probability in this selected topic. This stochastic process repeats for every word in the tweets of every user.

On the other hand, unlike a tweet created by one single user, a retweet may be forwarded by multiple users, and the retweet should exhibit the topics of interest to these forwarders. Therefore, to generate a word in a retweet, a topic would be first picked based on the topic distributions of all the users who forwarded this retweet. A word x would then be chosen from the word distribution specific to this picked topic.

Let us formally describe the URM model. Let y index each topic exhibited by words in tweets \mathbf{w} . As a result, there is a word distribution, denoted by ϕ_y , for each tweet topic y . To avoid manually setting the number of tweet topics, we assume ϕ_y itself to be a random variable drawn from a Dirichlet process. As discussed before, draws from a DP often share common values and thus naturally form clusters. Instead of being pre-specified, the number of clusters, which is often smaller than the total number of draws, varies with respect to data.

As a result, the global probability of generating tweets $p(\mathbf{w})$ is distributed as a DP, which can be expressed with a stick-breaking representation [94]:

$$p(\mathbf{w}) = \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k}, \quad (4.1)$$

where ϕ_k follows the prior H over multinomial distributions: $\phi_k \sim H$; δ_{ϕ} is a probability measure concentrated at ϕ ; and $\beta = (\beta_k)_{k=1}^{\infty} \sim \text{GEM}(\alpha)$ is an infinite

sequence defined as:

$$\beta'_k \sim \text{Beta}(1, \alpha), \quad \beta_k = \beta'_k \prod_{l=1}^{k-1} (1 - \beta'_l)$$

The global distribution defined in Equation (4.1) captures the homogeneity for the tweet behavior of users for the global population, but it does not reflect each individual user’s behavior. As stated earlier, to capture the heterogeneity among all users, we characterize each user by a mixture model. These mixture models of all users are linked together via the global distribution defined in Equation (4.1). Linking these mixture models is significant and useful in that it allows the tweet topics to be shared among all users. For instance, consider a user who is interested in the food topic and the politics topic, and another user who likes the food topic and the technology topic. It would be helpful for a model to relate the food topic discovered in the analysis of the former user to that detected from the latter user.

Specifically, the probability of generating user u ’s tweets can be written as:

$$p(\mathbf{w}_u) = \sum_{k=1}^{\infty} \pi_{uk} \delta_{\phi_k}, \quad (4.2)$$

where the mixing proportion $\pi_u = (\pi_{uk})_{k=1}^{\infty} \sim DP(\lambda, \beta)$. In this way, we introduce another layer of DP for the mixture of tweet topics in each user.

Moreover, as discussed before, each retweet is modeled as a mixture of a set of topics as well. Let z index each topic exhibited by words in retweets \mathbf{x} . σ_z denotes the word distribution for retweet topic z . R_j denotes the set of all the users who forwarded the j -th retweet message. The probability of generating the j -th retweet is thus given as:

$$p(\mathbf{x}_j) = \sum_{k=1}^{\infty} \eta_{jk} \delta_{\sigma_k}. \quad (4.3)$$

where $\eta_j = (\eta_{jk})_{k=1}^{\infty} \sim DP(\mu, \frac{1}{|R_j|} \sum_{u \in R_j} \pi_u)$. As a result, the generation of a retweet is attributable to the topics of interest to all of its forwarders. The stick-breaking representation of the URM model is depicted in Figure 4.1(a).

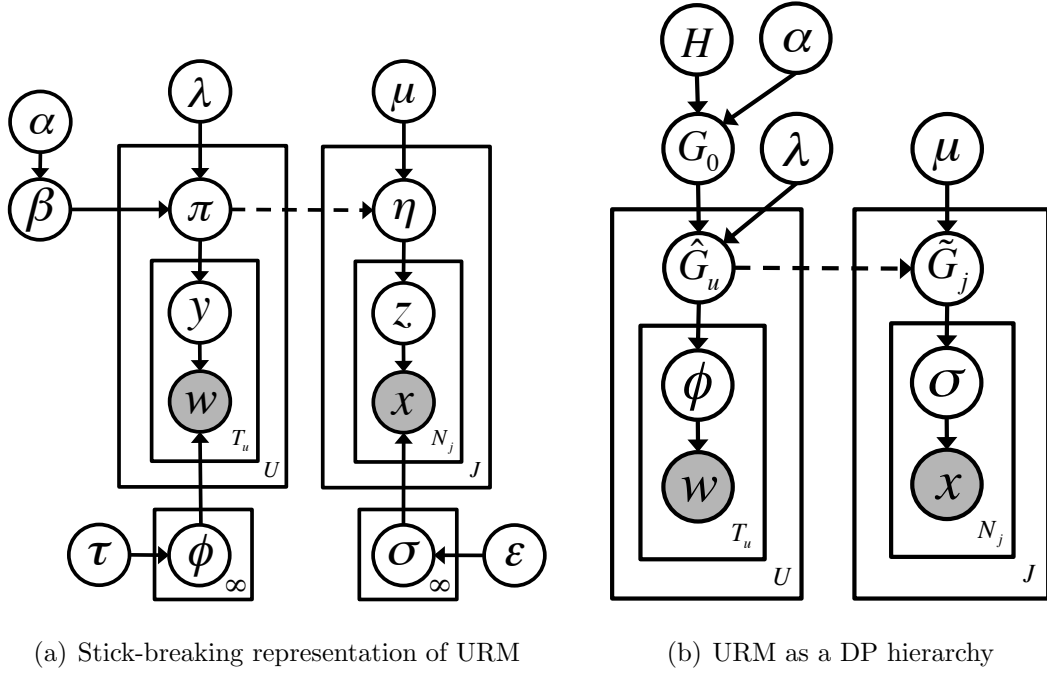


Figure 4.1: Graphical models for URM

4.2.2 URM as a Three-layer DP Hierarchy

In a way, URM generalizes HDP by using a three-layer Dirichlet process hierarchy for retweet modeling. The URM model defines a set of random probability measures in each layer of the DP hierarchy. In particular, first we draw a global probability measure G_0 from a DP with base measure H and concentration parameter α influencing the sparsity of the global topic distribution:

$$G_0 \sim DP(\alpha, H). \quad (4.4)$$

To characterize personal topical interest in tweeting, we then draw a topic distribution \hat{G}_u from the global probability measure over the topic space G_0 for each user:

$$\hat{G}_u \sim DP(\lambda, G_0) \quad (4.5)$$

with concentration parameter λ .

To model the generation of retweets, since a single message can be retweeted

by multiple users, for each tweet we draw a probability measure \tilde{G}_j from a set of multiple topic probability measures, $\{\hat{G}_u|u \in R_j\}$, corresponding to all the forwarders of this tweet, R_j .

Here we introduce a novel notion of drawing a probability measure from a set of probability measures. An equivalent representation of the set of probability measures $\{\hat{G}_u|u \in R_j\}$ is given by a DP with base measure $\frac{1}{|R_j|} \sum_{u \in R_j} \hat{G}_u$ which averages the probability measures in this set. We show that a DP with an average of multiple probability measures as its base measure is equivalent to a standard DP in the following. Suppose $\sigma_1, \dots, \sigma_{i-1}$ are observed samples from \tilde{G}_j . The probability of the i -th draw σ_i to be sampled from \tilde{G}_j can then be given by integrating out \tilde{G}_j using the properties of the Dirichlet distributed partitions [81] and replacing the base measure with the average of multiple probability measures:

$$\begin{aligned} \sigma_i|\sigma_1, \dots, \sigma_{i-1}, \mu, \tilde{G}_j &\sim \frac{1}{i-1+\mu} \sum_{k=1}^{i-1} \delta_{\sigma_k} \\ &+ \frac{\mu}{|R_j|(i-1+\mu)} \sum_{u \in R_j} \hat{G}_u, \end{aligned} \quad (4.6)$$

which gives a standard Dirichlet process. The URM model as a DP hierarchy is illustrated in Figure 4.1(b).

4.2.3 Bayesian Inference for URM

To estimate the latent topic structures in URM, we perform posterior inference to “invert” the generative process described above. In particular, we develop an efficient Markov chain Monte Carlo (MCMC) algorithm [93], or more precisely a Gibbs sampler, to approximate the posterior for URM. In a Gibbs sampler, each latent variable is iteratively sampled conditioned on the observations and all the other latent variables, so the key to Gibbs sampling is to derive a full conditional distribution for each latent variable, which is given in the following.

Sampling \mathbf{y} :

Let w_{ua} denote the a -th word in user u 's tweets. Given the current values of the remainder of the variables, denoted by \bullet , the probability of word w_{ua} assigned to an existing topic k can be derived as:

$$p(y_{ua} = k | \bullet) \propto (c_{uk}^{-(ua)} + \lambda\beta_k) \frac{e_{kw_{ua}}^{-(ua)} + \tau_{w_{ua}}}{e_{k*}^{-(ua)} + \tau_*}, \quad (4.7)$$

whereas the probability that the topic assignment y_{ua} takes on a new value k_{new} is given by:

$$p(y_{ua} = k_{\text{new}} | \bullet) \propto \frac{\lambda\beta_{k_{\text{new}}}}{V}, \quad (4.8)$$

where $c_{uk}^{-(ua)}$ denotes the number of words in user u 's tweets assigned to topic k , excluding the current assignment y_{ua} . $e_{kw}^{-(ua)}$ denotes the number of times word w is assigned to topic k across all tweets, excluding the current assignment. V is the total number of unique words in the vocabulary.

During the sampling process, if a topic assignment takes on a new value k_{new} , we include this new topic $\phi_{k_{\text{new}}}$ into the set of tweet topics, for which we draw a new global proportion $\beta_{k_{\text{new}}}$. On the other hand, if, as a result of updating topic assignments, none of words is assigned to some topic, we delete this unallocated topic from the set of tweet topics, and update the global proportions β accordingly.

Sampling \mathbf{z} :

Gibbs sampling for retweet topics \mathbf{z} is similar to that for tweet topics \mathbf{y} . Let x_{jb} denote the b -th word in the j -th retweet. The probability of word x_{jb} assigned to a previously used topic k can then be given by:

$$p(z_{jb} = k | \bullet) \propto (d_{jk}^{-(jb)} + \sum_{u \in R_j} \mu\pi_{uk}) \frac{g_{kx_{jb}}^{-(jb)} + \epsilon_{x_{jb}}}{g_{k*}^{-(jb)} + \epsilon_*}, \quad (4.9)$$

while the probability that the topic assignment z_{jb} takes on a new value k_{new} is as follows:

$$p(z_{jb} = k_{\text{new}} | \bullet) \propto \frac{\sum_{u \in R_j} \mu\pi_{uk_{\text{new}}}}{V}, \quad (4.10)$$

where $d_{jk}^{-(jb)}$ denotes the number of words in the j -th retweet assigned to topic k , excluding the current assignment z_{jb} . $g_{kx}^{-(jb)}$ denotes the number of times word x

is assigned to topic k across all retweets, excluding the current assignment. R_j denotes the set of all the users who forwarded the j -th retweet.

Sampling β :

Following the simulation of new tables in the CRF introduced in [89], the prior global proportions β can be sampled by simulating how new topics are created for c_{uk} draws from the DP with precision $\lambda\beta_k$ (dishes in the CRF), which is a sequence of Bernoulli trials for each u and k :

$$p(m_{ukr} = 1) = \frac{\lambda\beta_k}{\lambda\beta_k + r - 1} \quad \forall r \in [1, c_{uk}]. \quad (4.11)$$

A posterior sample of β is then obtained by:

$$\beta \sim \text{Dirichlet}(m_1, \dots, m_K, \alpha), \quad (4.12)$$

where $m_k = \sum_u \sum_r m_{ukr}$, and K is the number of active topics with which there exist words associated. β has dimension $K + 1$ because the mass for α in the Dirichlet distribution corresponds to generating a new topic out of an infinite set of empty topics. If a topic has lost all its words, it is merged with the unknown topics in the mass associated with α . Iterative sampling based on Equations (4.11) and (4.12) gives the posterior samples of β , which are needed by sampling tweet topics \mathbf{y} .

Sampling π :

Similarly, Equations (4.9) and (4.10) for sampling retweet topics \mathbf{z} require the posterior samples of π . The posterior proportion π_u for user u is given by:

$$\pi_u \sim \text{Dirichlet}(n_{1u}, \dots, n_{Ku}, \lambda), \quad (4.13)$$

where $n_{ku} = \sum_j \sum_r n_{jkur}$. The auxiliary Bernoulli variable n_{jkur} for retweet j , topic k and user u is defined as:

$$p(n_{jkur} = 1) = \frac{\mu\pi_{uk}}{\mu\pi_{uk} + r - 1} \quad \forall r \in [1, d_{jk}]. \quad (4.14)$$

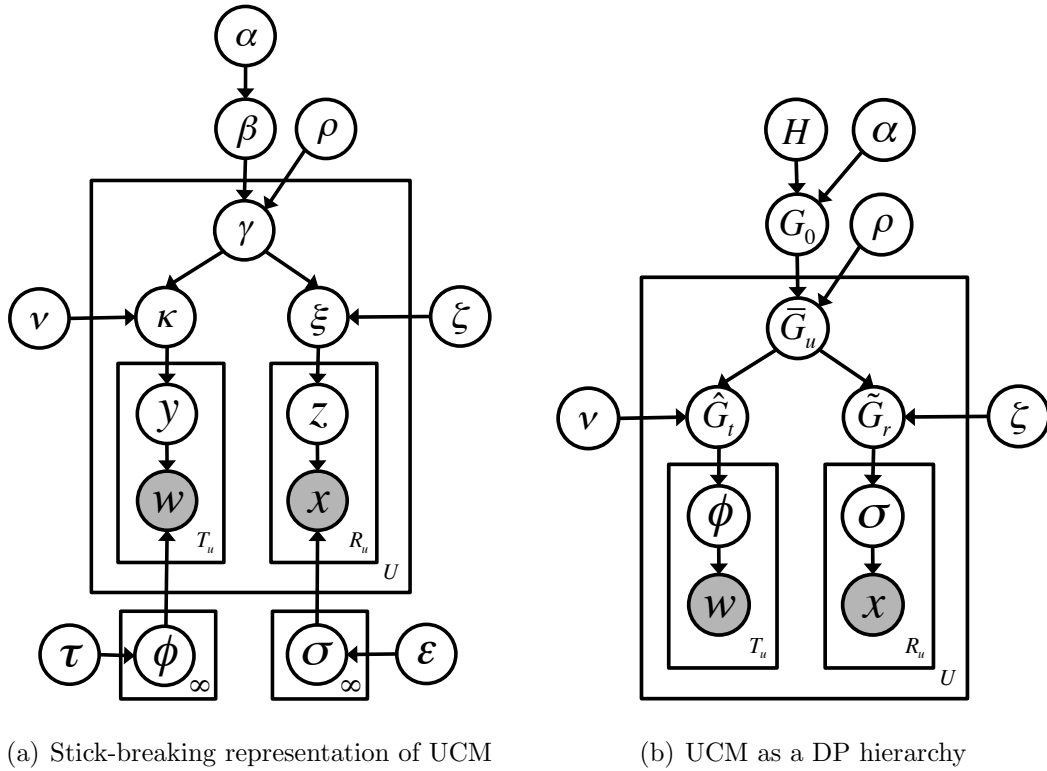


Figure 4.2: Graphical models for UCM

All the above posterior distributions create a Markov chain for Gibbs sampling. The Gibbs sampler for URM iteratively samples \mathbf{y} , \mathbf{z} , β , and π as described above in turn. Since the nearly samples from a Markov chain are usually correlated with each other, we only kept the samples from every five iterations to collect independent samples. Moreover, we discarded the samples in the burn-in period (the first 20% of samples).

4.3 User-centric Model (UCM)

4.3.1 Generative Process for UCM

The User-Retweet Model characterizes each user and each retweet as a unique mixture model. In other words, it constructs a separate mixture model for each retweet in addition to user modeling. Given that users' behavior of both tweet

and retweet reflects their distinct preference and topical interest, an alternative to user modeling would be introducing a random measure specific to each user that captures his or her unique interest. There often exist differences between the tweet interest and the retweet interest of a user. For instance, a user may be interested in retweeting jokes, but he or she could never tweet anything joking. To differentiate a user’s interest in tweet and retweet, we should introduce two random measures which capture his or her tweet interest and retweet interest, respectively. This alternative model is referred to as User-centric Model (UCM).

Formally, in the UCM model, we introduce a probability measure \bar{G}_u specific to any user u , which is distributed as a DP:

$$\bar{G}_u \sim DP(\rho, G_0), \quad (4.15)$$

where $G_0 \sim DP(\alpha, H)$. Equation (4.15) can be represented with a stick-breaking process as:

$$\bar{G}_u = \sum_{k=1}^{\infty} \gamma_{uk} \delta_{\phi_k}, \quad (4.16)$$

where $\gamma_u = (\gamma_{uk})_{k=1}^{\infty} \sim DP(\rho, \beta)$. The mixing proportion γ_u quantifies the user u ’s common interest in each different topic, which reflects the homogeneity of u ’s behavior of tweet and retweet. To separate the modeling of tweet interest and that of retweet interest, we draw from \bar{G}_u a probability measure \hat{G}_t for tweet generation and a probability measure \tilde{G}_r for retweet generation:

$$\hat{G}_t \sim DP(\nu, \bar{G}_u), \quad (4.17)$$

$$\tilde{G}_r \sim DP(\zeta, \bar{G}_u). \quad (4.18)$$

Using the stick-breaking representation, Equations (4.17) and (4.18) can be expressed as:

$$\hat{G}_t = \sum_{k=1}^{\infty} \kappa_{uk} \delta_{\phi_k}, \quad (4.19)$$

$$\tilde{G}_r = \sum_{k=1}^{\infty} \xi_{uk} \delta_{\sigma_k}, \quad (4.20)$$

where $\kappa_{uk} = (\kappa_{uk})_{k=1}^{\infty} \sim DP(\nu, \gamma)$, which measures the user u 's topical interest in tweet, and $\xi_{uk} = (\xi_{uk})_{k=1}^{\infty} \sim DP(\zeta, \gamma)$, which quantifies u 's retweet interest over the topics. The stick-breaking representation of UCM is illustrated in Figure 4.2(a). Figure 4.2(b) depicts the graphical model for UCM as a DP hierarchy.

4.3.2 Bayesian Inference for UCM

We develop a Gibbs sampler specifically for Bayesian inference for UCM, which is similar to the sampler for URM. In this section, we describe the posterior distributions for topic assignments \mathbf{y} and \mathbf{z} , conditioned on the values of all the other variables.

Sampling \mathbf{y} :

The Gibbs sampling equation for topic assignment y_{ua} of the a -th word in user u 's tweets is:

$$p(y_{ua} = k | \bullet) \propto (c_{uk}^{-(ua)} + \nu\gamma_{uk}) \frac{e_{kw_{ua}}^{-(ua)} + \tau_{w_{ua}}}{e_{k*}^{-(ua)} + \tau_*}, \quad (4.21)$$

whereas a new value k_{new} is sampled for y_{ua} based on the following probability:

$$p(y_{ua} = k_{\text{new}} | \bullet) \propto \frac{\nu\gamma_{uk_{\text{new}}}}{V}, \quad (4.22)$$

where $c_{uk}^{-(ua)}$ denotes the number of words in user u 's tweets assigned to topic k , excluding the current assignment y_{ua} , and $e_{kw}^{-(ua)}$ denotes the number of times word w is assigned to topic k across all tweets, excluding the current assignment.

Sampling \mathbf{z} :

For the b -th word in user u 's retweets, a previously seen topic k is sampled from the distribution given by:

$$p(z_{ub} = k | \bullet) \propto (f_{uk}^{-(ub)} + \zeta\gamma_{uk}) \frac{g_{kx_{ub}}^{-(ub)} + \epsilon_{x_{ub}}}{g_{k*}^{-(ub)} + \epsilon_*}, \quad (4.23)$$

whereas the probability that the topic assignment z_{ub} takes on a new value k_{new} is:

$$p(z_{ub} = k_{\text{new}} | \bullet) \propto \frac{\zeta\gamma_{uk_{\text{new}}}}{V} \quad (4.24)$$

where $f_{uk}^{-(ub)}$ denotes the number of words in user u 's retweets assigned to topic k , excluding the current assignment z_{ub} , and $g_{kx}^{-(ub)}$ denotes the number of times word x is assigned to topic k across all retweets, excluding the current assignment.

4.4 Empirical Evaluation

To evaluate the quality of our proposed models, URM and UCM, we conducted experiments on a real-world dataset crawled from Twitter. First, we demonstrate the latent topics discovered by both models, which qualitatively reflect the effectiveness of the models. Then, we quantitatively measure the quality of the topics discovered by our proposed models and the baseline. Finally, we assess and compare the predictive power and generalizability of these models to objectively evaluate their effectiveness.

4.4.1 Dataset and Experiment Settings

Our experiments were conducted on a Twitter dataset collected between October 2009 and January 2010. This dataset was crawled based on the follow network in a breadth-first search manner. Users' tweet content and retweet activities were collected during the crawling process. We used the tokenizer from the TweetNLP project [47] in order to improve the accuracy of the recognized terms in the noisy text. Furthermore, we reduced the inherent noise of tweets, by removing terms that appear in less than 20 tweets.

URM and UCM require a set of hyper-parameters to be determined a priori. In our experiments, we set the hyper-parameters: $\alpha = 1, \lambda = 0.5, \mu = 0.5, \tau = 0.1, \epsilon = 0.1, \rho = 0.5, \nu = 0.5, \zeta = 0.5$. We ran the Gibbs sampling algorithms for 1000 iterations.

4.4.2 Topics Produced by URM and UCM

Given URM and UCM as Bayesian nonparametric models, both models are able to automatically determine the number of latent topics based on the data. To estimate the posterior over the number of topics, during the Gibbs sampling process we collected posterior samples after the Markov chain had converged. The plots in Figure 4.3 depict the histograms of the number of tweet/retweet topics produced by URM and UCM. From the histograms, it is seen that both models discovered $100 \sim 120$ topics from tweets/retweets. Since the uncovered latent topics reflect the effectiveness of URM and UCM, and provide insights about users' interest on Twitter, we will illustrate a sample of distilled latent topics later in this section.

A latent topic can be represented as a distribution over a fixed set of words in the vocabulary. For a tweet topic k , the posterior distribution of words can be calculated as:

$$\phi_{kw} = p(w|y = k) = \frac{e_{kw} + \tau_w}{\sum_{w=1}^V (e_{kw} + \tau_w)}, \quad (4.25)$$

where the counter e_{kw} gives the number of times word w is assigned to topic k across all tweets. Similarly, the posterior distribution of words for a retweet topic k can be computed as:

$$\sigma_{kx} = p(x|z = k) = \frac{g_{kx} + \epsilon_x}{\sum_{x=1}^V (g_{kx} + \epsilon_x)}, \quad (4.26)$$

where the counter g_{kx} gives the number of times word x is assigned to topic k across all retweets. Since the number of latent topics might vary during the Gibbs sampling process, we collected samples when the Markov chain had converged to a stationary distribution.

Table 4.1 shows a sample of latent topics produced by URM and UCM in some run of Gibbs sampling. Every topic is represented by the set of top five most probable words under this topic. Intuitively, it is clear that both models distilled meaningful topics from tweets and retweets. For example, the first row in

Table 4.1: A sample of latent topics produced by URM and UCM

Model	Topic	Top-5 words
URM	Tweet	music, album, band, play, show
		love, kids, mom, fun, baby
		real, estate, property, read, home
	Retweet	travel, hotel, flight, new, italy
		social, media, twitter, facebook, marketing
		book, read, amazon, writing, author
UCM	Tweet	god, jesus, lord, church, his
		video, music, live, album, show
		green, car, energy, hybrid, carbon
	Retweet	google, iphone, apple, ipad, app
		film, movie, avatar, tv, trailer
		bowl, super, nfl, football, sports

this table, which lists words *music*, *album*, *band*, *play*, and *show*, indicates a music-related topic, and the topic given in the first row for UCM, which is represented by *god*, *jesus*, *lord*, *church*, and *his*, is clearly relevant to Christianity. Naturally, such anecdotal evidence is very hard to generalize. In the next section, we will present a quantitative measure to evaluate the quality of the distilled topics.

4.4.3 Topic Quality

We followed the *word intrusion* approach introduced in [30] to quantify the topic quality: In the word intrusion task, to evaluate the quality of a topic, the subject was presented with six randomly order words, which consisted of the five words with the highest probability under the topic and a word from another topic from the same model. The task of the user was to find the word which was out of place or did not belong with the others, i.e., the *intruder*. In case of semantically coherent topic words, the intruder should be easily found. To further test the

interaction between latent topics, the intruder was chosen from a set of words which had a low probability (out of the top 25 words) in the evaluated topic and a high probability (top 5 of the remaining words) in another topic.

Let j_k^m denote the index of the intruder among the words generated from topic k distilled by model m . Further let i_{ks}^m denote the intruder selected by subject s on the set of words generated from topic k distilled by model m , and let S denote the number of subjects. According to [30], the model precision on topic k is defined by the fraction of subjects that agree with the model on the topic:

$$\text{MP}_k^m = \sum_{s=1}^S \mathbb{1}(i_{ks}^m = j_k^m) / S. \quad (4.27)$$

The precision of model m computes the average of MP_k^m over all K evaluated topics: $\text{MP}^m = \sum_{k=1}^K \text{MP}_k^m / K$.

We compared the results of URM and UCM with those of Hierarchical Dirichlet Processes (HDP), which is a different Bayesian nonparametric model. In HDP, the words of each user are generated from a unique probability measure, which is drawn from a DP. The probability measures for all users share the same base measure, which is a draw from another DP. More details of HDP can be found in Section 2.4. In our experiments, we built three independent HDPs as baselines based on different pieces of the data. One of the HDPs, which we refer to as HDP-t, was run on the top of tweet text, while neglecting the information of the retweet structure. In other words, HDP-t considers the words in each users' tweets only to be generated from a user-specific probability measure. In contrast, another HDP, referred to as HDP-r, did the opposite by running on words in users' retweets without taking their tweets into account. The last HDP, which we refer to as HDP-tr, integrated information of tweets and retweets by aggregating the words from both tweets and retweets of each user, which were considered to be generated from a user-specific probability measure.

We computed overall model precisions for the three baselines HDP-t, HDP-r

Table 4.2: Comparison of model precisions

HDP-t	HDP-r	HDP-tr	URM	UCM
0.643	0.628	0.654	0.688	0.751

Table 4.3: Model precisions of URM and UCM over tweet/retweet topics

Topic	URM	UCM
Tweet topic	0.718	0.769
Retweet topic	0.640	0.731

and HDP-tr, as well as our models URM and UCM. As shown in Table 4.2, HDP-tr performed better than both HDP-t and HDP-r, suggesting that integrating the content of tweets and retweets in a model produces higher-quality topics than separate modeling of tweets and retweets. Our models URM and UCM outperformed all the three baselines, which clearly demonstrates the capability of the proposed models to distill high-quality latent topics. Specifically, UCM gave a higher model precision than URM. To track the cause of the performance difference, we computed model precisions of URM and UCM over tweet topics and retweet topics separately. From Table 4.3, it is observed that UCM is superior to URM in terms of quality of both tweet topics and retweet topics. Moreover, UCM gave a much higher model precision than URM over retweet topics, which implies that it should be more appropriate to have one \tilde{G}_r for each user than having one \tilde{G}_j for each retweet, since the user-specific \tilde{G}_r would have sufficient content from the user to characterize his or her retweet interest. The difference in modeling the retweet structure also improves the tweet topic quality of UCM over that of URM.

4.4.4 Predictive Power Analysis

As generative models, URM, UCM and HDP are all able to generate and predict unseen new data. We evaluated the predictive power and generalizability of both models using the standard *perplexity* metric [23]. The perplexity is monotonically decreasing in the likelihood of the unseen test data. Hence, a lower perplexity score indicates stronger predictive power. Formally, the perplexity is defined as:

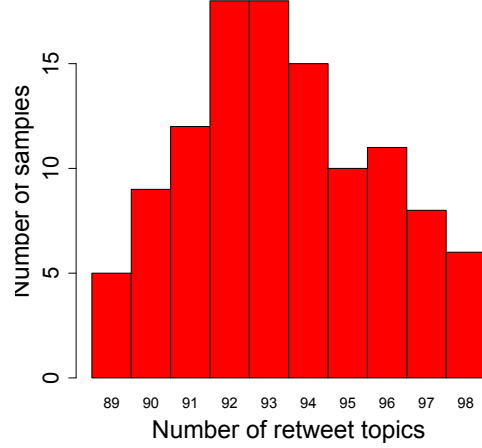
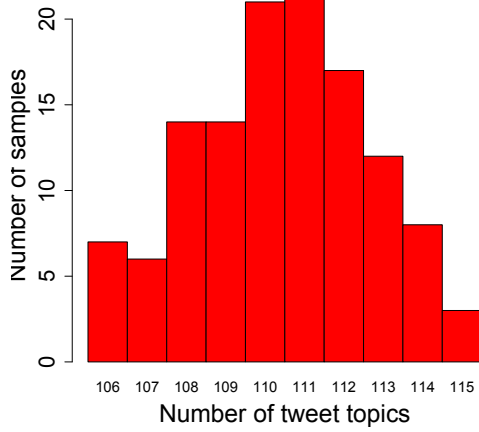
$$\text{perplexity}(D_{\text{test}}) = \exp \left\{ -\frac{\sum_{u \in D_{\text{test}}} \log p(\mathbf{w}_u)}{\sum_{u \in D_{\text{test}}} |\mathbf{w}_u|} \right\}, \quad (4.28)$$

where D_{test} denotes the test set of all Twitter users' words in tweets/retweets. To calculate the word perplexity, we held out 20% of the data D_{test} for test purposes and trained the models on the remaining 80%.

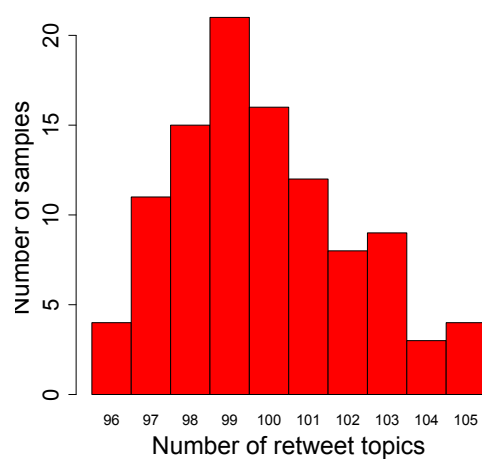
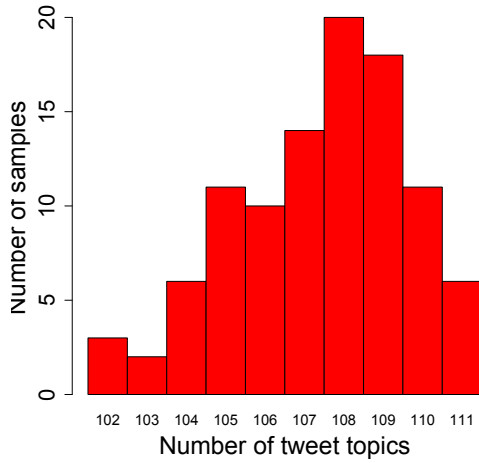
Figure 4.4 compares the word perplexity for HDP, URM and UCM. For our models URM and UCM, we calculated perplexity on the words in tweets as well as perplexity on the words in retweets. Since HDP-t and HDP-r applied to tweets and retweets, respectively, we calculated perplexity for HDP-t on the words in tweets and perplexity for HDP-r on the words in retweets. From this figure, we see that UCM gave the lowest perplexity on both tweets and retweets, confirming its strongest predictive power and the best generalizability. Although URM is inferior to UCM, the URM model outperformed the two HDP models in generating and predicting the words in both tweets and retweets. We also calculated overall perplexity on both tweets and retweets for HDP-tr, URM and UCM. As a result, UCM gave the lowest overall perplexity of 1540.6. The second-best URM had overall perplexity of 1723.1, which outperformed HDP-tr with overall perplexity of 1779.3. The experimental results are consistent with the results of the evaluation of topic quality. It validates the hypothesis that appropriate modeling of the retweet structure enhances the effectiveness of the model.

4.4.5 Conclusion

In this work, we proposed two Bayesian nonparametric models, URM and UCM, to analyze microblog data. Both models do not require the number of topics as an input parameter. Instead, they automatically determine the number of topics based on the observed microblog data. URM and UCM not only are able to integrate the analysis of tweet content and that of retweet behavior of users in the same statistical framework, but also jointly model users' interest in tweet and retweet. We devised two collapsed Gibbs samplers to estimate the latent topic structures in the two models, respectively. Thorough experiments on real-world microblog data were conducted to investigate the quality and the predictive power of both models.



(a) Histogram of the number of tweet topics for URM (b) Histogram of the number of retweet topics for URM



(c) Histogram of the number of tweet topics for UCM (d) Histogram of the number of retweet topics for UCM

Figure 4.3: Histogram of the number of latent topics produced during the Gibbs sampling process

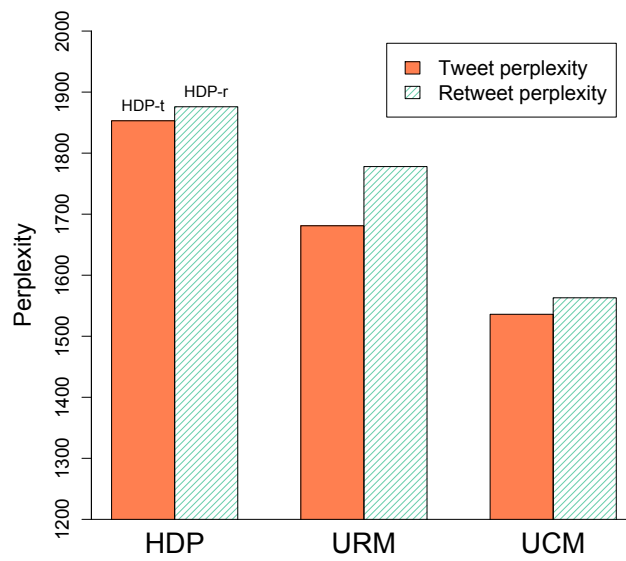


Figure 4.4: Comparison of word perplexity for HDP, URM and UCM

CHAPTER 5

Topic-specific Authority Analysis on Content Sharing Services

5.1 Introduction

Over the last decade, we have been witnessing the explosion of Web 2.0 applications. In the new era of Web 2.0, web users are participating not only as passive consumers of content provided by websites, but also as contributors creating content collaboratively with fellow users, commonly referred to as *user-generated content*. With the rapid growth of Web 2.0, a variety of *content sharing services*, such as *Flickr*¹, *YouTube*², *Blogger*³, and *TripAdvisor*⁴ etc, have become tremendously popular over the recent years. These websites enable users to create and share with each other various kinds of *resources*, such as photos, videos, and travel blogs, etc.

The sheer amount of user-generated content made available by the content sharing services can be both a blessing and a curse. From the point of user modeling, richer information content helps to build more accurate user profiles, leading to better services for consumers. On the other hand, the vast quantity of user-generated content available can often complicate the decision making process, as consumers do not have the time or ability to examine all data or compare

¹<http://www.flickr.com>

²<http://www.youtube.com>

³<http://www.blogger.com>

⁴<http://www.tripadvisor.com>

all options [11]. On a content sharing website, the overwhelming resources vary greatly in quality, which result in confusion, sub-optimum decisions or dissatisfaction with choices made by users [96]. Therefore, it is highly significant to develop a principled method that identifies a set of authorities, who created quality-assured resources, from a massive number of contributors of content.

A lot of work has been done on authority identification in the context of social network and web structure analysis. However, most of these studies, such as typical PageRank, only infer *global* authoritativeness of each user, without assessing the authoritativeness in a particular aspect of life (topics) [85, 69, 33, 107]. It does not make sense for a user to find global authorities on a content sharing website. After all, each user has unique topical interest. For example, on Flickr, a user who is interested in photographing sunsets may look for a photographer expert in this specific topic and learn from her photos about the skill of sunset photography. On the other hand, no one is an authority on every topic. Clearly, *topic-specific* authority analysis provides a more detailed authoritativeness portfolio for a user, which is critical for authority identification on content sharing services.

A common way of distilling latent topics is to build a probabilistic topic model on the usage data collected from a *sharing log*. In a content sharing website, a *sharing log* stores users' posting and tagging history, as illustrated by Figure 5.1 in Section 5.3. However, the sharing log does not contain any information about the content quality of resources, based on which authorities are identified. It would be counterintuitive to assume a high sharing frequency for every authority. Therefore, a data source in addition to the sharing log is clearly needed. Luckily, a *favorite log* made available by a content sharing website provides a valuable signal for the derivation of the content quality of resources. On current content sharing services, a resource is often presented with a *favorite* button, which a user clicks if he or she likes the resource. A *favorite click* represents an endorsement of the content quality of the resource by the user. The favorite log records the set

of favorite clicks as user feedback, as illustrated by Figure 5.2 in Section 5.3.

Despite considerable research on the sharing log for various applications, little is known about the emerging favorite log. It is nontrivial to leverage a favorite log for topic-specific authority analysis in that users do not explicitly specify their topical motivates under the favorite clicks. A statistical model, built upon both the sharing log and the favorite log, is imperative to uncover each user’s authoritativeness on different topics.

In this work, we propose a novel Bayesian model to identify a list of authorities on given topic(s), which we refer to as Topic-specific Authority Analysis, abbreviated as TAA. The TAA model characterizes each user’s topical authoritativeness by introducing a user-specific random vector over latent topics. To assess the topical authoritativeness, TAA exploits favorite clicks through systematically modeling the associations among users’ interest and authoritativeness as well as the topics of favorited resources. We propose to learn the parameters in the TAA model from a training dataset of observations constructed from both usage logs. To this end, a novel logistic likelihood function specialized for the training set is proposed to relate the parameters to the observations. Bayesian inference for a model with a logistic likelihood has long been recognized as a hard problem. We extend typical collapsed Gibbs sampling by introducing auxiliary variables to overcome this problem. With the inferred parameters, an analysis framework is introduced to produce an ordered list of topic-specific authorities by their authoritativeness degrees that satisfy the user’s query intent.

The major contributions of our work are summarized as follows:

1. We propose a novel Bayesian model, TAA, to address the new problem of topic-specific authority analysis on content sharing services by jointly leveraging the two data sources: *sharing log* and *favorite log*.
2. We propose a principled approach to training dataset construction, in which

a novel logistic likelihood function is introduced.

3. We extend classic collapsed Gibbs sampling by *data augmentation* to infer the parameters in the TAA model with the new logistic likelihood.
4. We conducted thorough experiments on the datasets collected from two specific real-world content sharing websites. Experimental results confirm the effectiveness of TAA in topic-specific authority identification as well as the predictive power of the TAA generative model.

5.2 Related Work

Much work has been done on authority identification based on a network structure. The two most representative studies are PageRank [85] and HITS [69]. Zhang *et al.* [119] tested PageRank and HITS on a specific online community for expert identification. Jurczyk and Agichtein [64] employed the HITS algorithm to discover authorities in question answer communities. Kempe *et al.* [67] abstracted authority analysis into a influence maximization problem and pioneered the Linear Threshold (LT) Model and Independent Cascade (IC) Model to explain the spread of influence in a social network. Along with subsequent works, such as [33] and [72], all these methods are only after the identification of global authorities instead of authorities for specific topics. Although Barbieri *et al.* [9] extended the LT and IC models to be topic-aware, the topics are obtained based on the network structure, while totally neglecting valuable textual content.

A few studies have been conducted to find topic-level authorities in the context of structure analysis of the web graph and social networks. Given the popularity of PageRank, it is only natural to extend it for topical authority analysis. Topic-Sensitive PageRank (TSPR) [54] was such an extension that computes per-topic PageRank scores for webpages. TSPR biases the computation of PageRank

by replacing the classic PageRank’s uniform teleport vector with topic-specific ones. However, it requires an existing manually categorized topic hierarchies to derive per-topic teleport vectors. In [100], Tang *et al.* proposed a Topical Affinity Propagation (TAP) model for topic-level social influence. But, similar to TSPR, TAP requires a separate preprocess to obtain a set of topics. TwitterRank [114] extended TSPR to find topic-level influencers on Twitter. Instead of predefined topic hierarchies, a set of topics is first produced by typical LDA [23] on the tweets. Then TwitterRank applies a method similar to TSPR to compute the per-topic influence rank. Nallapati *et al.* proposed Link-PLSA-LDA [80] on a hyperlink network to estimate the influence of blogs. These studies differ from our TAA model in that they do not exploit the valuable favorite signal to model topic-specific authoritativeness. Although TwitterRank and Link-PLSA-LDA applied to the settings different from ours, we adapted them to the authority identification on content sharing services by building proper graph structures, and compared them with our TAA in empirical studies.

There also exist a few pieces of prior work on finding important users in various applications. Chen *et al.* [31] proposed a latent factor model for rating prediction, based on which reputable users are identified. Zhao *et al.* [120] found topic-level experts on community question answering services, and recommended appropriate experts to answer new questions. In [20], Fellowship-LDA was proposed to identify topic-specific influencers on microblogs. All these methods find important users under different contexts, with the data different from ours in nature.

In the context of recommender systems, a few topic modeling studies related to our work have been conducted. Several latent factor models were proposed for tag recommendation on social media [16, 18, 14]. Wang and Blei [106] developed the collaborative topic regression (CTR) model to recommend scientific articles to users of an online community. Agarwal and Chen [2] proposed fLDA, which is a new matrix factorization method integrating LDA priors, to predict ratings


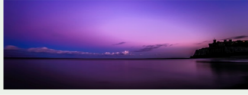
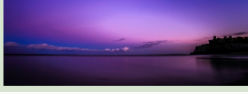


User ID	Tag	Photo
14829	kingsgate castle	
14829	broadstairs	
14829	beach	
319526	wareham forest	
319526	sunrise	
...

Figure 5.1: Sample records from the sharing log of a photo sharing website

in recommender system applications. Despite the relevance of these studies to our work, there are clear differences between them. To make recommendations, CTR utilizes scalar rating responses different from the binary favorite feedback exploited by TAA. fLDA is able to take binary responses, but it aims to predict scalar ratings of users on various items, which is different from the ultimate goal of our work.

5.3 Problem Statement

In a nutshell, the objective of this work is developing a statistical model that identifies the authorities on a content sharing website specific to given query topic(s). A topic-specific authority is defined as a user who excels in the specified topic. For example, given *city lights* as a query topic on a photo sharing website, the topic-specific authority model is intended to retrieve a list of users who are expert in city lights shooting at night.

A content sharing website generally logs a massive number of posting and tagging records that reflect every user’s unique interest and taste. These records constitute a *sharing log* that a content sharing service keeps track of. Figure 5.1 presents a few sample records from the sharing log of a photo sharing website. Each row in the table represents a record indicating that user u assigned tag t to resource r (i.e., a photo) which was posted by herself. For notational convenience, let L denote the total number of unique users in the log, M_u denote the number of resources posted by user u , and N_r denote the number of tags assigned to resource r . The notations used throughout this chapter are given in Table 7.1. Some of the notations will be explained in later sections.

A feasible solution to topic-specific authority identification is adapting the classic topic model *Latent Dirichlet Allocation* (LDA) [23] to historical data in the sharing log. Specifically, we employ typical LDA on the sharing data by regarding a user as a document in a corpus, a tag as a word in a document. By fitting the topic model to observational data collected from the sharing log, we infer the optimal values of parameters θ and φ . The probabilities θ (i.e., $p(z|u)$) give the topic distribution for each user, and the probabilities φ (i.e., $p(t|z)$) give the tag distribution for each topic. As a result, topic-specific authorities can be derived from the distributions $p(z|u)$ and $p(t|z)$ by the standard *query likelihood model*, where each user is scored by the likelihood of generating a given query. In particular, given a set of tags as a query q , we compute the likelihood $p(q|u)$ for each user u by:

$$p(q|u) = \prod_{t \in q} p(t|u) = \prod_{t \in q} \sum_{z=1}^K p(t|z)p(z|u). \quad (5.1)$$

The users with the highest likelihood $p(q|u)$ are then identified as topic-specific authorities.

The LDA-based authority analysis exploits the fact that a user is interested in a particular topic if he or she frequently labels photos with the tags specific to this

Table 5.1: Notations used throughout this chapter

Notation	Description
u	User identity
t	Tag identity
r	Resource identity
z	Topic assignment of a tag
f	Binary favorite feedback
L	Total number of unique users
K	Total number of unique topics
R	Total number of unique resources
V	Total number of unique tags in the vocabulary
M_u	Number of resources posted by user u
N_r	Number of tags assigned to resource r
N_u	Number of tags assigned by user u
θ	Per-user topic distribution
φ	Per-topic tag distribution
α, β	Dirichlet priors on Multinomial distributions
η	Per-user topical authoritativeness

topic. It further assumes that the more frequently a user uses the tags covering a specific topic, the more authoritative he or she should be on this topic. However, this is an arguable assumption which is not always valid. Tagging frequently on a particular topic does not automatically imply that the user is an authority on this topic. In fact, an authority does not have to tag more than the other users on the topic he or she excels in. For example, on a travel blogging service, a blogger who posts a number of articles tagged with *London travel* may not be an authority on blogging about traveling London, given the unknown quality of these articles. It is likely that he or she is new to blogging, in which case the articles could be at a beginner level in quality. On the other hand, an actual authority may post only a couple of blogs about London travel, but he or she can specialize in this specific topic, leading to the favorable high-quality blogs. By analyzing the usage data






User ID	Favorited Photo
82310	
185963	
28737	
49856	
93274	
...	...

Figure 5.2: Sample records from the favorite log of a photo sharing website

from a real-world sharing log, we observed that users’ tag frequency is actually independent of their authoritativeness.

Since the sharing log reports posting and tagging information, but we are looking for the information about the content quality of posted resources, a supplementary data source is needed. Fortunately, a *favorite log* available in most of the content sharing services should help to infer the content quality of resources. A favorite log consists of the records of each user’s favorites. Figure 5.2 depicts a few sample records from the favorite log of a photo sharing website. Each row in the table represents a record indicating that user u added resource r (i.e., a photo) to his or her favorites. A *favorite click* can be interpreted as the user’s vote in favor of the content quality of the favorited resource. It motivates our modeling the favorite signal to infer the content quality of resources based on which topic-specific authorities are identified.

As discussed above, users’ topical interest and topical authoritativeness have different implications. A favorite log enables us to separate the analysis of users’

topical authoritativeness from that of their topical interest. In order to jointly model the two factors, we need to construct a Bayesian model which specifies a generative process much more complex than that of typical LDA. The Bayesian model is intended to exploit both the sharing signal and the favorite signal by leveraging the two usage logs.

Problem Statement. *Given the usage data collected from a sharing log and a favorite log, we aim to design a stochastic process that simulates how the data is generated, based on which a generative model is developed to identify authorities specific to given query topic(s) on a content sharing service.*

5.4 Topic-specific Authority Analysis

Naturally, no one is an authority on every topic, which implies that each user’s authoritative degrees should be evaluated specific to individual topics. Moreover, users’ topical authoritativenesses are different from each other. Therefore, in our proposed TAA model, we introduce a K -dimensional random vector over topics to characterize topical authoritativeness. The random vector is designed to be specific to individual user u , denoted by $\boldsymbol{\eta}_u$, meaning that each user has a unique topical authoritativeness. An entry of random vector $\boldsymbol{\eta}_u$ is a latent variable η_{uz} reflecting user u ’s authoritative degrees on topic z . We assume that $\boldsymbol{\eta}_u$ is generated from a K -dimensional Multivariate Gaussian distribution:

$$\boldsymbol{\eta}_u \sim \text{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{5.2}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean vector and the covariance matrix, respectively. We choose the Multivariate Gaussian distribution due to its nice invariance property as a prior distribution. As will be discussed later, Multivariate Gaussian is a conjugate prior of our likelihood function, meaning that the posterior distribution of $\boldsymbol{\eta}_u$ will also be a Multivariate Gaussian. This trick benefits inference for our TAA model by computational convenience. The values of $\boldsymbol{\eta}_u$ for each user will be

learned from the usage data collected from the sharing log as well as the favorite log.

A *favorite click* reflects a positive feedback from the user on the content quality of the specific resource. Therefore, to represent a favorite feedback, we introduce a binary random variable specific to individual user u and individual resource r , denoted by f_{ur} . The binary variable f_{ur} takes value 1 if the user u favorited the particular resource r , 0 otherwise. Introducing f_{ur} helps to relate the resource r to the user u who favorited r . More precisely, user u favorites resource r ($f_{ur}=1$), if the topical authoritativeness of r 's owner exhibited by the resource r matches with u 's topical interest. For instance, a user, who is interested in photos of Yellowstone National Park, may favorite the Yellowstone photos from a photographer who is expert in taking shots for Yellowstone National Park. On the other hand, user u does not favorite resource r ($f_{ur}=0$), if u 's interest and the authoritativeness of r 's owner exhibited by the resource r fall into different sets of topics. For example, a user, who is interested in blogs about Yellowstone travel, is unlikely to favorite the low-quality articles from a blogger who is new to this particular topic.

Since the topical motivate under each favorite click is hidden and unavailable directly, we need to identify the topics in which a user is interested as well as the topics on which a user is authoritative. To this end, we propose a novel generative model on the usage data for topic distillation. With the distilled topics, we specify the likelihood of a favorite feedback f_{ur} from user u on r with the logistic function by:

$$p(f_{ur} = 1 | \boldsymbol{\eta}_{u'}, \hat{\mathbf{z}}_u, \hat{\mathbf{z}}_{u'r}) = \frac{1}{1 + e^{-\boldsymbol{\eta}_{u'}^\top (\hat{\mathbf{z}}_u \circ \hat{\mathbf{z}}_{u'r})}} \quad (5.3)$$

$$p(f_{ur} = 0 | \boldsymbol{\eta}_{u'}, \hat{\mathbf{z}}_u, \hat{\mathbf{z}}_{u'r}) = 1 - \frac{1}{1 + e^{-\boldsymbol{\eta}_{u'}^\top (\hat{\mathbf{z}}_u \circ \hat{\mathbf{z}}_{u'r})}} \quad (5.4)$$

where u' denotes the user who posted resource r (i.e., r 's owner); $\hat{\mathbf{z}}_u$ denotes the topic distribution for user u 's interest; $\hat{\mathbf{z}}_{u'r}$ denotes the topic distribution for the resource r posted by user u' , and \circ denotes the Hadamard (element-wise)

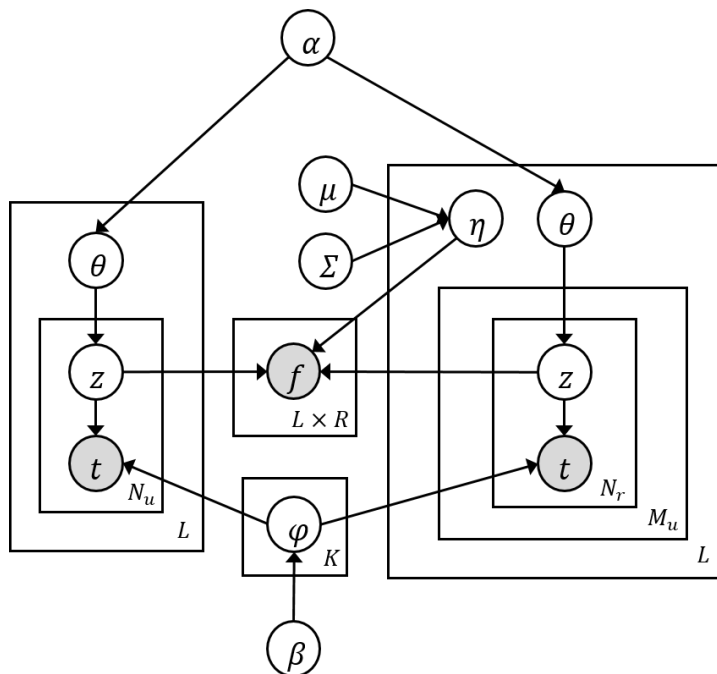


Figure 5.3: Graphical model for Topic-specific Authority Analysis

product. The element-wise product of $\hat{\mathbf{z}}_u$ and $\hat{\mathbf{z}}_{u'r}$ captures similarity between the topic distributions for the resource r and the interest of the user u who favorited r , which is parameterized by the owner u' 's topical authoritativeness $\boldsymbol{\eta}_{u'}$. If the topic distribution for user u 's interest is similar to the one for resource r , there should be a specific set of topics prominent in both u 's interest and resource r . A favorite click $f_{ur} = 1$ then indicates that this specific set of topics are the ones that the resource r 's owner u' is expert in, and thus should be parameterized by high authoritativeness degrees. In this way, we uncover the hidden topical motivate under each favorite click.

Figure 5.3 shows the graphical model for our TAA, with the notations described in Table 7.1. The generative process of a user's tags and favorite feedback is summarized in Figure 5.4. A favorite feedback is naturally associated with a tuple (u, r) , where r denotes a resource, and u denotes the user who favorited r . To obtain individual user u 's interest distribution over topics, each user is viewed

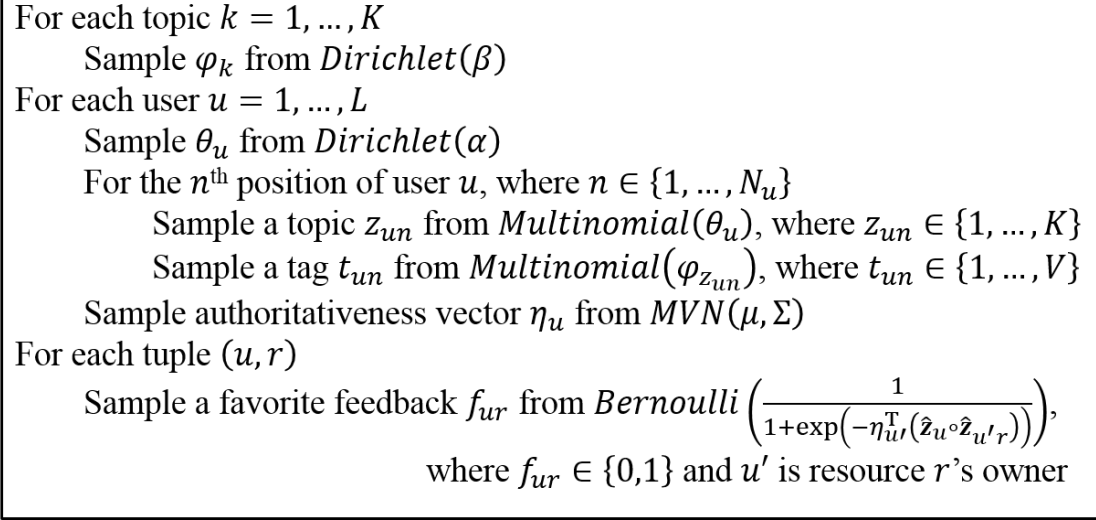


Figure 5.4: Generative process for Topic-specific Authority Analysis

as a mixture of topics from which tags are drawn. More specifically, for each user $u \in \{1, \dots, L\}$, we first pick a topic distribution θ_u from a Dirichlet prior with parameter α . Then, to generate the n^{th} tag in the resources posted by u , a topic z_{un} is sampled from θ_u , after which the tag t_{un} is drawn from the tag distribution $\varphi_{z_{un}}$ for topic z_{un} . With all the obtained topics, we compute individual user u 's topical interest distribution $\hat{\mathbf{z}}_u$ by aggregating u 's topic assignments. On the other hand, the topic distribution $\hat{\mathbf{z}}_{u'r}$ for individual resource r posted by user u' is obtained in a similar way, except that $\hat{\mathbf{z}}_{u'r}$ is computed by counting u' 's topic assignments specific to resource r only.

The topic distributions $\hat{\mathbf{z}}_u$ and $\hat{\mathbf{z}}_{u'r}$ enable the generation of favorite feedback. In particular, for each tuple (u, r) , the binary favorite feedback f_{ur} is sampled from a Bernoulli distribution with parameter $\frac{1}{1+e^{-\eta_u^T(\hat{\mathbf{z}}_u \circ \hat{\mathbf{z}}_{u'r})}}$. More specifically, we compute the likelihoods of $f_{ur} = 1$ and $f_{ur} = 0$ using Equation (5.3) and Equation (5.4), respectively. As a result, $f_{ur} \in \{0, 1\}$ is drawn from a Bernoulli distribution of the two likelihoods.

The various parameters we can learn from TAA characterize the different factors that affect the model structure. For a user u , the K -dimensional vector $\boldsymbol{\eta}_u$

quantifies u 's unique authoritativeness over topics, and the value θ_{uz} gives the probability that u is interested in topic z . For a topic z , the value φ_{zt} indicates the probability of tag t belonging to topic z . The inferred quantities serve as the inputs to our authority analysis framework, which will be described later.

5.5 Inference for TAA

In this section, we present how the parameters of the TAA model are inferred from the usage data collected from the sharing log and the favorite log. More specifically, we first construct a training dataset from the usage data, with which a new Bernoulli likelihood parametrized by a logistic function is specified. Finally, an extension of traditional Gibbs sampling specialized for the logistic likelihood function is proposed to infer the optimal values of the parameters.

5.5.1 Preference Learning

We learn the parameters of the TAA model from a training set of observations constructed from the usage data. As mentioned above, the favorite log consists of user preferences for resources in a content sharing service. One important fact about the favorite log is that only positive observations are available – each favorite click is viewed as positive feedback for the corresponding tuple (u, r) , i.e., $f_{ur} = 1$. However, there are not such clear conclusions for $f_{ur} = 0$. Considering the non-clicked tuples (u, r) (i.e., user u did not click on the *favorite* button for resource r .) as negative feedback ($f_{ur} = 0$) would misinterpret the signal of these tuples, since there are actually at least two different interpretations for any non-clicked tuple. One possibility is a negative feedback, meaning that the user did not like the resource and did not want to add it to his or her favorites. Another possibility is a missing value, indicating that the user did not even see the resource, in which case whether the user favorited the resource is unknown.

On the other hand, the non-clicked tuples should not be simply ignored, as typical machine learning models are not able to learn anything from the positive observations alone. To overcome the problem of missing negative feedback ($f_{ur} = 0$), we use tuple pairs as training data instead of individual tuples. As opposed to treating non-clicked tuples as negative observations, we assume that users prefer the resources, for which they clicked on the *favorite* buttons, over the other non-clicked resources from the same owner. More specifically, suppose that r_i and r_j represent two resources posted by a user. Given two tuples (u, r_i) and (u, r_j) , user u prefers r_i over r_j if and only if r_i was favorited by u while r_j was not, which is denoted by $r_i \succ_u r_j$. Formally, we create training data \mathcal{D} by including the pairwise preference relations as follows:

$$\mathcal{D} = \{(u, r_i, r_j) | r_i \succ_u r_j\}, \quad (5.5)$$

where each preference relation $o = (u, r_i, r_j)$ is a training sample representing the fact that user u prefers r_i over r_j . For the resources that are both favorited by a user, we cannot infer any preference. The same is true for two resources either of which a user did not favorite.

As discussed above, we construct the observational dataset \mathcal{D} using the induced preference relations in place of the raw favorite feedback f_{ur} . As a result, the likelihood functions (5.3) and (5.4) need to be extended to incorporate the pairwise preference. Therefore, we reformulate the likelihood of a preference relation as:

$$p(r_i \succ_u r_j | \boldsymbol{\eta}_{u'}, \hat{\mathbf{z}}_u, \hat{\mathbf{z}}_{u'r_i}, \hat{\mathbf{z}}_{u'r_j}) = \frac{1}{1 + e^{-\boldsymbol{\eta}_{u'}^\top (\hat{\mathbf{z}}_u \circ \hat{\mathbf{z}}_{u'r_i} - \hat{\mathbf{z}}_u \circ \hat{\mathbf{z}}_{u'r_j})}}. \quad (5.6)$$

The probability $p(r_i \succ_u r_j | \boldsymbol{\eta}_{u'}, \hat{\mathbf{z}}_u, \hat{\mathbf{z}}_{u'r_i}, \hat{\mathbf{z}}_{u'r_j})$ gives the likelihood that user u prefers resource r_i over resource r_j , both owned by user u' . Let Θ denote the set of parameters of the TAA model. The likelihood of observing all the preference relations in training data \mathcal{D} is then given by:

$$p(\mathcal{D} | \Theta) = \prod_{(u, r_i, r_j) \in \mathcal{D}} \frac{1}{1 + e^{-\boldsymbol{\eta}_{u'}^\top (\hat{\mathbf{z}}_u \circ \hat{\mathbf{z}}_{u'r_i} - \hat{\mathbf{z}}_u \circ \hat{\mathbf{z}}_{u'r_j})}}. \quad (5.7)$$

5.5.2 Bayesian Inference

Typical LDA-like generative models employ collapsed Gibbs sampling to infer their parameters [52, 55, 90]. However, Bayesian inference for a model with the logistic likelihood function (7.4) has long been recognized as a hard problem, due to the analytically inconvenient form of the Gibbs sampler for a logistic likelihood [57, 43, 51]. In this section, we present an extension of traditional collapsed Gibbs sampling to infer the parameters in TAA. Our algorithm takes advantage of the *data-augmentation* idea by introducing auxiliary variables to the posterior distribution. It extends the very recent work on inference for logistic models [88, 32] to learn a Bayesian model for topic-specific authority analysis. Specifically, using the ideas of introducing Pólya-Gamma variables presented in [88, 32], we are able to derive the posterior probabilities for the Gibbs sampler analytically. Part of the derivation is provided in the appendix.

Let us first familiarize ourselves with a new family of Pólya-Gamma distributions [88].

Definition. *A random variable X has a Pólya-Gamma distribution with parameters $b > 0$ and $c \in \mathcal{R}$, denoted by $X \sim PG(b, c)$, if*

$$X \stackrel{d}{=} \frac{1}{2\pi^2} \sum_{k=1}^{\infty} \frac{g_k}{(k - 1/2)^2 + c^2/(4\pi^2)}, \quad (5.8)$$

where the $g_k \sim \text{Gamma}(b, 1)$ are independent Gamma random variables; the notation $\stackrel{d}{=}$ denotes equality in distribution.

The Pólya-Gamma family has been carefully constructed to yield a simple Gibbs sampler for the Bayesian logistic model. Let $\delta_{ur_{ij}}$ denote a Pólya-Gamma variable specific to (u, r_i, r_j) . With the introduction of the auxiliary random variable $\delta_{ur_{ij}}$, the likelihood function (7.4) can be represented as mixtures of Gaussians

with respect to a Pólya-Gamma distribution, which is rewritten as:

$$\begin{aligned} & p(r_i \succ_u r_j | \boldsymbol{\eta}_{u'}, \hat{\mathbf{z}}_u, \hat{\mathbf{z}}_{u'r_i}, \hat{\mathbf{z}}_{u'r_j}) \\ &= \frac{1}{2} e^{\frac{\boldsymbol{\eta}_{u'}^\top \hat{\mathbf{z}}_{ur_{ij}}}{2}} \int_0^\infty e^{-\frac{\delta_{ur_{ij}} (\boldsymbol{\eta}_{u'}^\top \hat{\mathbf{z}}_{ur_{ij}})^2}{2}} p(\delta_{ur_{ij}} | 1, 0) d\delta_{ur_{ij}}, \end{aligned} \quad (5.9)$$

where $\hat{\mathbf{z}}_{ur_{ij}} = \hat{\mathbf{z}}_u \circ \hat{\mathbf{z}}_{u'r_i} - \hat{\mathbf{z}}_u \circ \hat{\mathbf{z}}_{u'r_j}$.

As a result, the collapsed posterior distribution of TAA augmented with the variables δ is given by:

$$\begin{aligned} & p(\mathbf{z}, \delta, \boldsymbol{\eta} | \mathbf{t}, \mathbf{o}, \alpha, \beta, \mu, \Sigma) \\ & \propto \prod_{u=1}^L \frac{\prod_{k=1}^K \Gamma(c_{ku} + \alpha_k)}{\Gamma(\sum_{k=1}^K c_{ku} + \alpha_k)} \times \prod_{k=1}^K \frac{\prod_{t=1}^V \Gamma(g_{kt} + \beta_t)}{\Gamma(\sum_{t=1}^V g_{kt} + \beta_t)} \\ & \quad \times p(\boldsymbol{\eta} | \mu, \Sigma) \prod_{(u, r_i, r_j) \in \mathcal{D}} e^{\frac{\boldsymbol{\eta}_{u'}^\top \hat{\mathbf{z}}_{ur_{ij}} - \delta_{ur_{ij}} (\boldsymbol{\eta}_{u'}^\top \hat{\mathbf{z}}_{ur_{ij}})^2}{2}} p(\delta_{ur_{ij}} | 1, 0) \end{aligned} \quad (5.10)$$

where c_{ku} is the number of user u 's tags assigned to topic k , and g_{kt} is the total number of times tag t is assigned to topic k over the dataset. The detailed derivation of Equation (5.10) is provided in the appendix.

The univariate conditionals for a Gibbs sampler are then given as follows. The notation \bullet represents all the variables other than the one to be sampled.

$[p(\boldsymbol{\eta}_x | \bullet)]:$

We impose a zero-mean isotropic Gaussian prior on the K -dimensional random vector $\boldsymbol{\eta}_x$ which characterizes user x 's topical authoritativeness:

$$p(\boldsymbol{\eta}_x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\sum_k \eta_{xk}^2}{2\sigma^2}}. \quad (5.11)$$

Thanks to the invariance property of the conjugate prior, the posterior distribution of $\boldsymbol{\eta}_x$ is also a Multivariate Gaussian:

$$\begin{aligned} p(\boldsymbol{\eta}_x | \bullet) & \propto p(\boldsymbol{\eta}_x) \prod_{r_i \in R(x) \wedge r_j \in R(x)} e^{\frac{\boldsymbol{\eta}_x^\top \hat{\mathbf{z}}_{ur_{ij}} - \delta_{ur_{ij}} (\boldsymbol{\eta}_x^\top \hat{\mathbf{z}}_{ur_{ij}})^2}{2}} \\ & = \text{MVN}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x) \end{aligned} \quad (5.12)$$

where $r_i \in R(x)$ represents that resource r_i is posted by user x . The posterior mean μ_x and posterior covariance Σ_x are given by:

$$\begin{aligned}\mu_x &= \Sigma_x \left(\sum_{r_i \in R(x) \wedge r_j \in R(x)} \frac{1}{2} \hat{\mathbf{z}}_{ur_{ij}} \right) \\ \Sigma_x &= \left(\frac{1}{\sigma^2} I + \sum_{r_i \in R(x) \wedge r_j \in R(x)} \delta_{ur_{ij}} \hat{\mathbf{z}}_{ur_{ij}} \hat{\mathbf{z}}_{ur_{ij}}^\top \right)^{-1}\end{aligned}$$

$[p(z_{un}|\bullet)]:$

The posterior distribution of \mathbf{z} is:

$$\begin{aligned}p(\mathbf{z}|\bullet) &\propto \prod_{u=1}^L \frac{\prod_{k=1}^K \Gamma(c_{ku} + \alpha_k)}{\Gamma(\sum_{k=1}^K c_{ku} + \alpha_k)} \times \prod_{k=1}^K \frac{\prod_{t=1}^V \Gamma(g_{kt} + \beta_t)}{\Gamma(\sum_{t=1}^V g_{kt} + \beta_t)} \\ &\times \prod_{(u,r_i,r_j) \in \mathcal{D}} e^{-\frac{\boldsymbol{\eta}_{u'}^\top \hat{\mathbf{z}}_{ur_{ij}} - \delta_{ur_{ij}} (\boldsymbol{\eta}_{u'}^\top \hat{\mathbf{z}}_{ur_{ij}})^2}{2}}\end{aligned}\quad (5.13)$$

The univariate conditional distribution of one variable z_{un} given all the other variables is then given by:

$$\begin{aligned}p(z_{un} = k|\bullet) &\propto \frac{(c_{ku}^{-(un)} + \alpha_k)(g_{kt_{un}}^{-(un)} + \beta_{t_{un}})}{\sum_{t=1}^V g_{kt}^{-(un)} + \sum_{t=1}^V \beta_t} \\ &\times \prod_{(u,r_i,r_j) \in \mathcal{D}} p(r_i \succ_u r_j | \boldsymbol{\eta}_{u'}, \mathbf{z}_{-(un)}, z_{un} = k)\end{aligned}\quad (5.14)$$

where $c_{ku}^{-(un)}$ bears the same meaning of c_{ku} only with the n th tag of user u excluded; similarly $g_{kt}^{-(un)}$ is defined in the same way as g_{kt} only without the count for the n th tag of user u , and $\mathbf{z}_{-(un)}$ denotes the topics for all tags except z_{un} .

$[p(\delta_{ur_{ij}}|\bullet)]:$

By definition, the posterior distribution of the auxiliary variable $\delta_{ur_{ij}}$ turns out to be a Pólya-Gamma distribution:

$$\begin{aligned}p(\delta_{ur_{ij}}|\bullet) &\propto e^{-\frac{\delta_{ur_{ij}} (\boldsymbol{\eta}_{u'}^\top \hat{\mathbf{z}}_{ur_{ij}})^2}{2}} p(\delta_{ur_{ij}}|1, 0) \\ &= \text{PG}(1, \boldsymbol{\eta}_{u'}^\top \hat{\mathbf{z}}_{ur_{ij}})\end{aligned}\quad (5.15)$$

The above posterior univariate distributions create a Markov chain for Gibbs sampling. It has been shown that the stationary distribution of the Markov chain is just the sought-after posterior joint distribution [44]. Specifically, the Gibbs sampler iteratively draws samples from $p(\boldsymbol{\eta}_x|\bullet)$, $p(z_{un}|\bullet)$ and $p(\delta_{ur_{ij}}|\bullet)$ using Equations (5.12), (5.14) and (5.15), respectively. After the Gibbs sampler has run for an appropriate number of iterations (until the chain has converged to a stationary distribution), we draw a sample $\boldsymbol{\eta}_x$ for each user x , which quantifies x 's topical authoritativeness, and obtain the estimates for the distributions θ and φ via the following equations:

$$\theta_{uz} = \frac{c_{zu} + \alpha_z}{\sum_{k=1}^K c_{ku} + \sum_{k=1}^K \alpha_k} \quad (5.16)$$

$$\varphi_{zt} = \frac{g_{zt} + \beta_t}{\sum_{t=1}^V g_{zt} + \sum_{t=1}^V \beta_t} \quad (5.17)$$

5.5.3 Authority Analysis Framework

With the inferred parameters, we introduce an analysis framework for topic-specific authority identification. The analysis framework allows a user to issue a query q reflecting the topic(s) on which authorities are to be identified. The query q consists of a list of tags, where multi-occurrences of a tag are allowed to reflect its importance to the query topic(s). The analysis framework subsequently produces an ordered list of authorities by their authoritativeness degrees that satisfy the user's query intent.

To rank a list of authorities, the analysis framework requires **(a)** every user's topical authoritativeness: η , and **(b)** the topic(s) of query q : \mathbf{z}_q . When the TAA model is used as the underlying topic-specific authority analysis method, the topical authoritativeness η is produced as part of the results. To derive q 's topic(s) \mathbf{z}_q , we use the folding-in technique on TAA by treating the query as a new user, and perform the sampling for only the tags of the pseudo user. Given the derived topical authoritativeness $\boldsymbol{\eta}_u$ and the query topic(s) \mathbf{z}_q , we obtain the

final authoritativeness $\Psi(u, q) = \sum_{i=1}^{N_q} \eta_{uz_{qi}}$ for a user u with respect to the query q , where N_q denotes the number of tags in q . Finally, the users are returned in decreasing order of their authoritativeness $\Psi(u, q)$.

5.6 Empirical evaluation

In this section, we report the experimental results of the TAA model on real-world data collected from two specific content sharing services: *Flickr*⁵ and *500px*⁶. We quantitatively compare the results of TAA with those of several competitors on both datasets. We also give real examples of Flickr authorities identified by TAA. Analysis and discussion of the experimental results are presented in this section.

5.6.1 Data Collections

Although TAA is a generic Bayesian model which is applicable to topic-specific authority identification on various kinds of content sharing services, we conduct experiments on the real-world datasets collected from two specific websites *Flickr* and *500px* to evaluate the quality of identified authorities. *Flickr* is one of the most popular photo sharing website, which allows users to store, share, tag and organize their photos. The huge number of Flickr users calls for an topic-specific authority model to identify the best photographers for a specified query topic. As opposed to Flickr’s general user base, *500px* is a photo sharing platform catered to professional photographers. A distinct feature of 500px is the *Editors’ Choice* page⁷ which shows the finest photos hand-picked by the professional editors employed by 500px. These high-quality photos are used to derive the ground truth for our empirical evaluation.

We collected the sharing logs and the favorite logs from both Flickr and 500px.

⁵<http://www.flickr.com>

⁶<http://www.500px.com>

⁷<http://www.500px.com/editors>

Table 5.2: Statistics of Experimental Datasets

Data	#users	#photos	#tag asgmts	#fav. clicks
Flickr	21,054	204,335	3,014,813	1,562,805
500px	33,581	318,906	3,520,179	1,837,049

The usage data obtained from the collected logs were processed to create training data \mathcal{D} , on which a TAA model was built. Extra usage information was collected to derive the ground truth for both datasets, which will be described in the next subsection. The basic statistics of the Flickr dataset and the 500px dataset are given in Table 7.2.

5.6.2 Evaluation Strategy

Quantitatively evaluating the quality of topic-specific authority analysis is a difficult task, since a content sharing service generally does not explicitly specify real authorities given a topic. Luckily, the abundant information embedded in the databases of Flickr and 500px helps to derive ground truth of topic-specific authorities.

Flickr has a large number of user-created groups that allow people who have similar interests to get together and share their photos reflecting these interests. Each of the groups is generally dedicated to a certain topic, such as food, animals, certain photo techniques, or creative commons, etc. Every group has one or more administrators which can be viewed as the real authorities specific to the group topic. On the other hand, 500px organizes photos by category, such as wedding, underwater, concert, or transportation, etc. We rank the users for each category according to their numbers of photos get selected by the editors by category. The ranked list of users for each category is instead viewed as ground truth, since unlike a Flickr group, a 500px category has no administrators specific to the category topic.

Given the different kinds of ground truth for Flickr and 500px, we used different evaluation metrics to measure the quality of the results from compared algorithms. Let Q denote a set of queries. For each query $q \in Q$, each algorithm returns an ordered list of users by their authoritativeness. For the Flickr dataset, we employed the standard Mean Reciprocal Rank (MRR). The Reciprocal Rank of a ranked list is the multiplicative inverse of the rank of the first hit in the list. The MRR score of an algorithm is the average reciprocal rank obtained by the ranked lists given by the algorithm with respect to the query set Q . Formally,

$$MRR = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{rank_q} \quad (5.18)$$

where $rank_q$ is the rank of the first real authority in the ranked list for query q . By definition, a higher MRR score indicates a better algorithm. For the 500px dataset, on the other hand, we employed the Spearman’s rank correlation coefficient to assess the correlation between ground truth and a ranked list of users given by each algorithm. The Spearman’s coefficient ρ_q for query q can take a range of values from -1 to +1 ($\rho_q < 0$ for a negative correlation, $\rho_q > 0$ for a positive correlation). The Spearman’s coefficient ρ of an algorithm is the average Spearman’s coefficient over the query set Q given by the algorithm. Formally,

$$\rho = \frac{1}{|Q|} \sum_{q \in Q} \rho_q \quad (5.19)$$

5.6.3 Quality of Authority Analysis

In our experiments, we evaluated the quality of the authorities identified by the six algorithms, *Most-tagged*, *LDA*, *Most-favorited*, *TwitterRank*, *Link-PLSA-LDA*, and *TAA*. Given a set of tags as a query, the *Most-tagged* approach first identifies relevant photos by lexical matches against the query tags. The number of relevant photos of each user is viewed as his or her authoritativeness degree, by which *Most-tagged* produces a ranked list of users as a final result. By contrast, *LDA* identifies relevant photos using probabilistic topic modeling [23]. As a result,

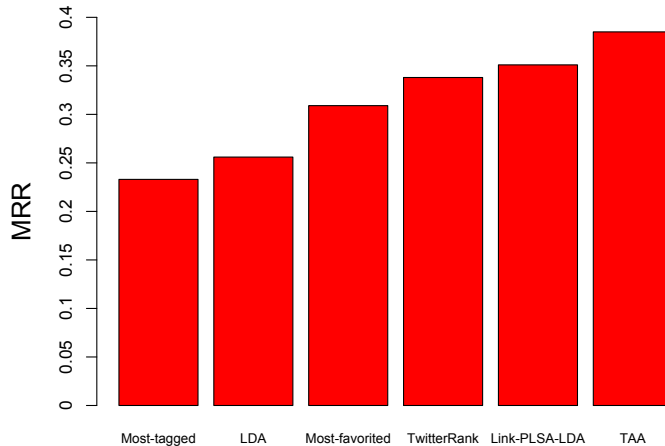


Figure 5.5: MRR for the Flickr dataset

users are ranked in descending order of the query likelihoods given by Equation (5.1). Note that both *Most-tagged* and *LDA* utilize observational data from the sharing log while neglecting the valuable signal from the favorite log. On the contrary, *Most-favorited* leverages both the sharing log and the favorite log in a way that produces an ordered list of users by the numbers of times their relevant photos are favorited. As opposed to the previous three approaches, *TwitterRank* and *Link-PLSA-LDA* both build upon the graph structure constructed from the favorite log. Specifically, we construct the graph by creating a node for each user. There exists a link from node u to node v if the user corresponding to u favorited any photo of the user corresponding to v . A user’s tags are associated with the corresponding node. The *TwitterRank* algorithm was originally proposed to find topic-level key influencers on Twitter [114]. It extends typical Topic-Sensitive PageRank [54] to compute per-topic influence scores. This requires a separate preprocess to create topics by running LDA on the text content associated with the nodes. The transition probability between two nodes in *TwitterRank* is defined based on the topical similarity between the corresponding users. Given the similar nature of the Twitter network and our constructed graph, we employ the *TwitterRank* algorithm to find topic-level authorities on a content sharing

service. On the other hand, *Link-PLSA-LDA* is a probabilistic topic model on a hyperlink/citation network, which jointly models text and citations to estimate the influence of blogs/publications [80]. We adapt it to our constructed graph for topic-specific authority analysis. In our experiments, for every topic-sensitive algorithm, we set the number of topics to 100. We set all symmetric priors as 0.1 for every model with Dirichlet priors. For our TAA, we ran Gibbs sampling for 500 iterations. These settings are fairly typical and their tuning is beyond the scope of this work.

To compute MRRs on the Flickr dataset, we randomly selected 200 Flickr groups, whose administrators were treated as the real authorities on the respective group topics. The *Top Tags* generated by Flickr for each group were fed as a query to each algorithm. Figure 5.5 shows the MRR score of each algorithm on the Flickr dataset. It is observed that *Most-tagged* and *LDA* were inferior to the other algorithms, as neither of them models the valuable favorite signal. On the contrary, by exploiting the favorite data, the algorithms *Most-favorited*, *TwitterRank*, *Link-PLSA-LDA* and the proposed *TAA* produced higher MRR scores. In particular, *TwitterRank* underperformed *Link-PLSA-LDA* and *TAA*, due to its separation between topic modeling and authority analysis. To further measure the improvement of *TAA* over the runner-up *Link-PLSA-LDA*, we performed a paired *t*-test between them, which gave p -value < 0.05 . It indicated that the improvement of *TAA* over *Link-PLSA-LDA* was *statistically significant*. This is not surprising because *Link-PLSA-LDA* as well as *TwitterRank* fail to uncover the latent topical motivate under each favorite click. Instead, they establish a link on the graph as long as a user favorited any photo of another, disregarding the identity of the photo as well as its underlying topics.

For 500px, we plot the Spearman’s coefficient for each algorithm in Figure 5.6. From this figure, we observe the pattern similar to that of Figure 5.5. *TAA* outperformed all the other algorithms, thanks to its unified framework of topic modeling

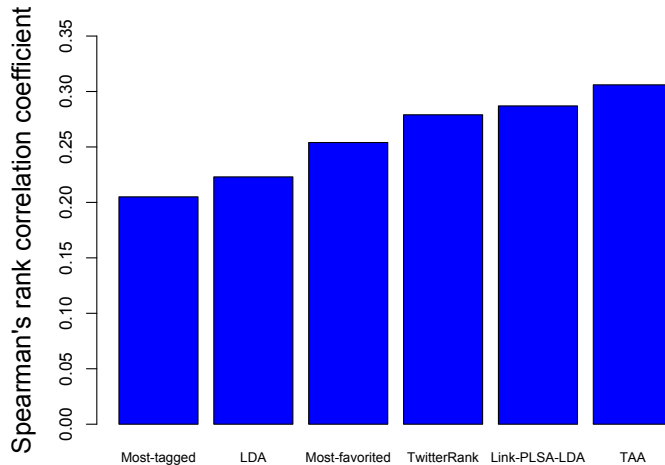


Figure 5.6: Spearman’s rank correlation coefficient for the 500px dataset

and authority analysis. In addition, *TAA* benefited from its ability to identify users’ topical authoritativeness by uncovering each favorite click’s underlying topical motivate and learning from pairwise resource preference.

5.6.4 Predictive Power Analysis

As generative models, our *TAA*, as shown in Figure 5.3, and the competitor *Link-PLSA-LDA* [80] are able to generate and predict unseen new data. We evaluated the predictive power and generalizability of both models using the standard *perplexity* metric [23]. The perplexity is monotonically decreasing in the likelihood of the unseen test data. Hence, a lower perplexity score indicates stronger predictive power. Formally, the perplexity is defined as:

$$perplexity(F_{\text{test}}) = \exp \left\{ -\frac{\sum_{f \in F_{\text{test}}} \log p(f)}{|F_{\text{test}}|} \right\}, \quad (5.20)$$

where F_{test} denotes the test set of favorites. For both Flickr and 500px, we held out 10% of the data for test purposes and trained the models on the remaining 90%.

Figure 5.7 and Figure 5.8 present the perplexity as a function of the numbers of topics for both models on Flickr data and 500px data, respectively. It is clear

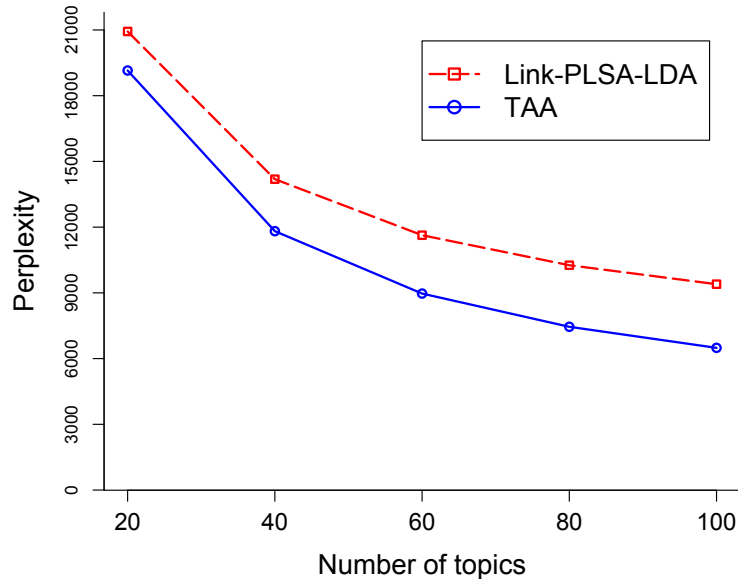


Figure 5.7: Perplexity for the Flickr dataset

that the *TAA* consistently produced lower perplexity scores than *Link-PLSA-LDA* for both Flickr and 500px, indicating that our *TAA* model has stronger predictive power and better generalizability. Moreover, *TAA* predicted unseen favorites even better as the number of topics increases.

5.6.5 Case Visualization

For the visualization of the *TAA* model, we performed searches on Flickr data for a list of photographers who are expert in two specific topics. Figure 5.9 shows the examples of photographers identified by *TAA* together with their ranks in the lists. To illustrate their expertise in photography, photos on the query topics are presented as well. For the first query topic: *winter snow landscape*, we see from the photos that the first user in the ranked list demonstrated the expertise in shooting snow landscape in winter. By contrast, the user in rank 100 seemed to have broader interests, not specializing in this specific topic. The last user looked

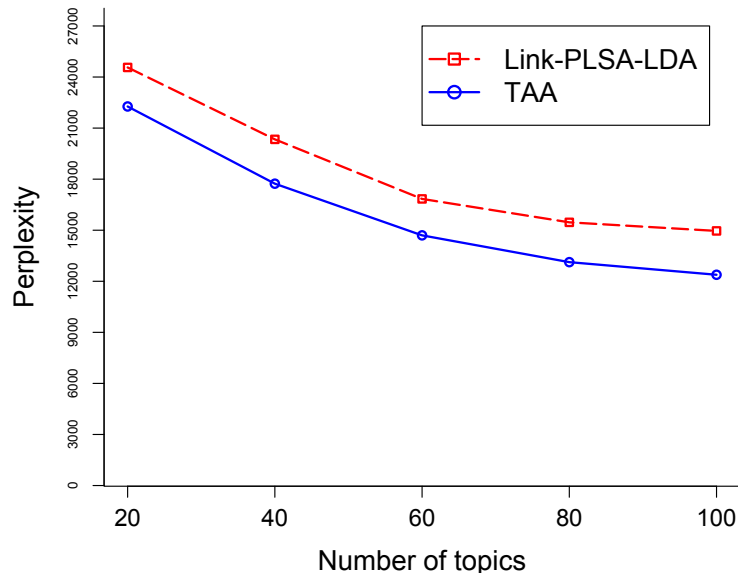





Figure 5.8: Perplexity for the 500px dataset

even irrelevant to the query topic. For the second query topic: *waterscape*, the user at the top was clearly superior to the others in waterscape shooting, although some photos from the last two users were somewhat related to the water topic.

5.7 Conclusion

This paper addresses the problem of authority analysis specific to given query topic(s) for users on a content sharing service. To model topic-specific authoritativeness, we introduce a novel method of Topic-specific Authority Analysis (TAA), which properly captures the associations among users’ interest and authoritative-ness as well as the topics of favorited resources to exploit the signal of favorite clicks. The parameters in the TAA model are learned from a training set of observations constructed from two data sources: *sharing log* and *favorite log*. To overcome the limitation of missing negative feedback, we propose a preference

Query topic: winter snow landscape		
User ID	Rank	Photos
29762217	Rank 1	
25355186	Rank 100	
11052010	Rank 1000	




Query topic: waterscape		
User ID	Rank	Photos
87620688	Rank 1	
25355186	Rank 100	
50701553	Rank 1000	

Figure 5.9: Examples of the ranked lists of photographers identified by *TAA* on Flickr data

learning technique embedding a new logistic likelihood function. An extension of typical collapsed Gibbs sampling is further proposed for Bayesian inference with the logistic likelihood. With the inferred parameters, our analysis framework produces a ranked list of authorities by their authoritativeness specific to given query topic(s).

We conducted thorough experiments on the datasets collected from two specific real-world content sharing websites, Flickr and 500px. Experimental results demonstrate that the TAA model outperforms the competitors, confirming its effectiveness in topic-specific authority analysis and its generalizability to unseen data.

Appendix

Let us derive the collapsed posterior distribution of TAA augmented with the variables δ , as follows:

$$\begin{aligned}
& p(\mathbf{z}, \delta, \eta | \mathbf{t}, \mathbf{o}, \alpha, \beta, \mu, \Sigma) \\
& \propto p(\mathbf{t}, \mathbf{z} | \alpha, \beta) p(\mathbf{o}, \delta | \mathbf{z}, \eta) p(\eta | \mu, \Sigma) \\
& = \int \int p(\mathbf{t}, \mathbf{z}, \theta, \varphi | \alpha, \beta) d\theta d\varphi \times p(\eta | \mu, \Sigma) p(\mathbf{o}, \delta | \mathbf{z}, \eta) \\
& = \int p(\mathbf{z} | \theta) p(\theta | \alpha) d\theta \times \int p(\mathbf{t} | \varphi, \mathbf{z}) p(\varphi | \beta) d\varphi \\
& \quad \times p(\eta | \mu, \Sigma) p(\mathbf{o}, \delta | \mathbf{z}, \eta) \\
& = \prod_{u=1}^L \int \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K \theta_{uk}^{\alpha_k - 1} \prod_{n=1}^{N_u} \theta_{uz_{un}} d\theta_u \\
& \quad \times \prod_{k=1}^K \int \frac{\Gamma(\sum_{t=1}^V \beta_t)}{\sum_{t=1}^V \Gamma(\beta_t)} \prod_{t=1}^V \varphi_{kt}^{\beta_t - 1} \prod_{u=1}^L \prod_{n=1}^{N_u} \varphi_{z_{un} t_{un}} d\varphi_k \\
& \quad \times p(\eta | \mu, \Sigma) p(\mathbf{o}, \delta | \mathbf{z}, \eta) \\
& \quad \text{(Expand out Dirichlet and Multinomial distributions)} \\
& = \prod_{u=1}^L \int \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K \theta_{uk}^{\alpha_k + c_{ku} - 1} d\theta_u \\
& \quad \times \prod_{k=1}^K \int \frac{\Gamma(\sum_{t=1}^V \beta_t)}{\sum_{t=1}^V \Gamma(\beta_t)} \prod_{t=1}^V \varphi_{kt}^{\beta_t + g_{kt} - 1} d\varphi_k \\
& \quad \times p(\eta | \mu, \Sigma) p(\mathbf{o}, \delta | \mathbf{z}, \eta) \\
& \propto \prod_{u=1}^L \frac{\prod_{k=1}^K \Gamma(c_{ku} + \alpha_k)}{\Gamma(\sum_{k=1}^K c_{ku} + \alpha_k)} \times \prod_{k=1}^K \frac{\prod_{t=1}^V \Gamma(g_{kt} + \beta_t)}{\Gamma(\sum_{t=1}^V g_{kt} + \beta_t)} \\
& \quad \times p(\eta | \mu, \Sigma) \prod_{(u, r_i, r_j) \in \mathcal{D}} e^{\frac{\boldsymbol{\eta}_u^\top \hat{\mathbf{z}}_{ur_{ij}} - \delta_{ur_{ij}} (\boldsymbol{\eta}_u^\top \hat{\mathbf{z}}_{ur_{ij}})^2}{2}} p(\delta_{ur_{ij}} | 1, 0)
\end{aligned}$$

CHAPTER 6

Inferring the Demographics of Search Users

6.1 Introduction

In recent years, we have been witnessing the rapid emergence of social networks and an increasing amount of user generated data. Meanwhile, it became apparent that the relevance of search results can be improved by personalization, i.e., by taking into account additional information about the user, such as interests, demographic and psychological traits, social background, or the context of the search. As a consequence, search engines have been evolving into *social-aware* platforms, Google’s social layer (Google+), and Bing’s *social pane* being perhaps the two most noteworthy examples.

While leveraging the background information about the users in ranking models has shown significant promise in enhancing users’ search experience both in academic [28] and industrial¹ studies, obtaining such features for all users can be difficult. For instance, a recent study suggests that only about 22% of Bing users are logged into Facebook account while searching², and even them may have not given the search engine access to their profile information. It would therefore be useful to be able to infer characteristics of users relevant to their search experience from information more readily available in the context of a search engine, such as the search query histories.

This work addresses the question of how demographic traits and users’ views

¹Google blog, <http://bit.ly/YaJvSm1>

²Search Engine Land: <http://se1nd.com/R6dpTN>

can be inferred based on the query histories. The main challenge, however, lies in the fact that only a very limited amount of data is available to allow training models for predicting such traits based on the search queries. For example, Microsoft user accounts provide no access to political views and religion, and only a small amount of data related to demographic traits.

How, then, can we build a machine learning system that predicts user demographics from query histories? What comes to the rescue is a substantial publicly available dataset called myPersonality³, offering psychometric test results and contents of the Facebook profiles for millions of anonymous Facebook users who volunteered to donate their data for research purposes. In particular, myPersonality database allows matching users' demographic profiles with their Facebook Likes, i.e., those online entities and Facebook Pages⁴ with which users have associated themselves using the Facebook Like button. Here we show how Facebook Likes can be used to build a model predicting users' individual traits that can be later applied to search query data.

There are two issues that need to be addressed to apply the model built on Facebook Likes to query histories. First, Facebook Likes need to be matched against queries. We achieve that by developing a common representation for Facebook Likes and search queries within the Open Directory Project (ODP)⁵ categories. Second, the distribution of users' traits differs between Facebook and search samples. Traditional machine learning algorithms commonly assume that the training and test samples are randomly drawn from the same distribution. To address this issue, we design a novel learner which is able to adapt the model learned from social data to search queries with a different distribution. This learner does not require unlabelled search queries to be seen at training time, which relaxes the condition of traditional transfer learning. Experimental results

³See <http://mypersonality.org/wiki> for more details.

⁴Facebook Pages, <http://www.facebook.com/pages>

⁵Open directory project, <http://www.dmoz.org>

show that the new learner can give high prediction accuracy as well as some interesting demographic results.

Hence, the work makes four important contributions:

- We show how to predict users' traits based on the search query logs by applying models developed on the Facebook Likes data.
- We show how to use ODP categories to match Facebook Likes with search queries.
- We demonstrate how to mitigate the problem of differing distributions of the traits.
- We provide experimental results that show the validity of the approach by comparing the predictions with ground truth data and with aggregate data at the US state level.

6.2 Related Work

Our work is related to a wide spectrum of previous studies ranging from inferring the demographics of individual users, to the application of user demographics in predicting global trends or individual behaviour.

The impact of demographics & personality Bachrach et al. [8] investigated the correlation between users' personality and the properties of their social network profiles. They showed that some personality traits such as Extroversion and Neuroticism can be accurately predicted based on the user's profile. A similar study was conducted by Quercia et al. [91] on Twitter users.

Kosinski et al. [71] demonstrated that there is a psychologically meaningful relationship between the users personality profiles obtained using a questionnaire, and their choice of websites extracted from Facebook Likes.

Weber and Castillo [108] used the Yahoo! query logs and profile information to compare the queries submitted by users with different demographics. They further analysed the queries submitted from each US ZIP code separately and mapped them against the US-census information for those area codes. Their results suggested that users with similar demographics are more likely to search for similar things. Weber and Jaimes [112] examined the queries submitted from different ZIP codes augmented by US-census data to highlight the differences in user behaviour and search patterns of various demographic groups. We take this line of previous work to the next level by showing that the demographics of users can be automatically predicted based on their past queries.

Lorigo et al. [74] discovered that male and female users have different search behaviour; for instance, females on average submit longer queries. Jansen and Solomon [61] found that males and females interact differently with sponsored search results.

Kharitonov and Serdyukov [68] demonstrated how reranking the search results based on users' genders may enhance their experience in particular for ambiguous queries.

Bennett et al. [12] inferred a compact density representation of locations of users that access different websites and showed that those features can be used for personalizing and reranking the search results.

Inferring user demographics Torres and Weber [102] reported that the reading levels of clicked pages are correlated with the demographic characteristics of the clicking users. Weber et al. [109, 110] relied on user clicks on political blogs annotated with *leaning* to assign a leaning score (left versus right) to queries.

Pennacchiotti and Popescu [87] used the linguistic content of user tweets, along with their other social features to predict the political orientation, ethnicity and the favourite business brands of Twitter users. They found the user-centric fea-

tures such as linguistic content to be more effective than social graph features in their classification task.

Ying et al. [115] showed that the users demographics can be predicted according to their mobile usage behaviour, such as the number of text messages sent or received. Otterbacher [84] inferred the author gender of IMDB reviews based on stylistic and content features.

Jones et al. [63] investigated the problem of inferring users demographics based on their queries but mostly focused on the privacy angle. They leveraged bag-of-word classifiers based on queries to train their models.

Perhaps in the most similar work to ours Hu et al. [58], predicted the users' ages and genders based on their browsing model. For each website in their corpus they used the Microsoft Live ID information of users that accessed them to build a demographic model. They then used these models to predict the ages and genders of other users that access the same website. In our approach we bring the social and query data into the same space but mapping them against the ODP categories. As a result, we have a much denser feature space that allows us to have high generalisability and cover several other interesting aspects such as religion and political views in the inference.

From query trends to global statistics Weber and Jaimes [111] monitored the Yahoo! query logs to determine if the same queries were submitted by different demographic groups at different times. Their analysis revealed that certain queries (e.g. movies) are searched by distinct demographics at different times, suggesting an *information flow* pattern between different groups of users.

Goel et al. [50] used query volume to predict the opening weekend box-office revenue of films, first-month sales of video games and the ranks of songs on the Billboard Hot 100 chart. In each of these cases, the authors found that there was a significant correlation between the query volume and future outcomes. Ettredge

et al. [40] performed a similar study but focused on predicting the unemployment rate.

Ginsberg et al. [48] accurately detected the influenza epidemics by only using the frequency and volume of certain queries in Google logs. Later on, Kong et al. [70] utilized click-through for the same purpose, and Culotta [36] repeated a similar analysis on Twitter data.

Domain adaptation and transfer learning Our work is also related to domain adaptation and transfer learning techniques. In domain adaptation [38] typically the same feature space is shared by the source and target domains. We also deal with two distinct source (social data) and target (queries) spaces in our experiments, and bridge them by mapping them to a single common space (based on ODP categories).

Transfer learning techniques can be used to resolve the problem of the different distributions between source and target spaces. It is worth noting that in contrast to typical transfer learning models [117, 41, 37, 7], our approach requires neither any data sharing between the source and target domains, nor any target data to be seen at training time.

6.3 Modeling User Demographics

As mentioned in the Introduction, we are addressing the problem of inferring users' traits from search queries based on the models trained on an independent set of Facebook Likes and profiles. We thus face two challenges

- How can we find a common representation for search queries and Facebook Likes?
- How can we address the problem that the users' traits are distributed dif-

ferently in those two datasets?

We address the first problem by mapping both search queries and Facebook Likes into a common representation given by the Open Directory categories, which form a mini-ontology of entities on the Web and can be thought of as a coarse grained representation of both search queries and Facebook Likes. Figure 6.1 illustrates the common representation based on the DMOZ Open Directory Project (ODP) categories. For Facebook Likes we turn the title of each *liked entity* into a query and submit it to a search engine (for example for the *lady gaga* Facebook Like, we submit the query *lady gaga*). We classify each of the top ten results returned by the search engine (Bing was used in this study) into one of the top two-levels of the DMOZ/ODP categories, assigning a maximum of three categories to each result. In total, there are 219 topical categories such as Arts/Movies, Business/Jobs and Computers/Internet. For learning the category classifiers we follow the approach described by [13] and apply logistic regression with L2 regularization on a 2008 crawl of the documents linked with the ODP index. Using the output of these classifiers we then represent each Facebook Like in the myPersonality dataset by a 219-dimensional vector. Each element of this vector denotes the number of times that a particular ODP category has been assigned to the search results returned for that Like. We then repeat the same process on search (Bing) users. To generate the topical feature vector for each user, we collect the queries from their search history and classify them in the same way as the we did for the Facebook Likes. Each user is represented again by a 219-dimensional vector, in which each element denotes the number of times the corresponding ODP category has been assigned to top-ranked documents returned for user queries. The feature values are normalized into probabilities so that they all sum to one for each user.

The second problem arises because of the differing users' traits distribution between users in the Facebook and search queries samples. Traditional machine learning algorithms commonly assume that the training data consist of samples



Figure 6.1: The workflow of our framework for inferring users’ demographics based on the search queries. On the left, the Facebook Likes of a small group of users are mapped to their corresponding ODP categories by issuing them as queries and classifying the top search results. On the right, the search users are represented similarly by the set of ODP categories associated with the top-ranked results returned for their queries.

randomly drawn from the same distribution as the test samples about which the learned model is expected to make predictions. This assumption is violated in our scenario where the model trained on Facebook data is applied to a query log to predict users’ demographic characteristics in the search engine. One of the examples is that there are relatively more female user in the Facebook (myPersonality) dataset, compared to search (Bing) users. Naively training on one dataset and testing on the other can significantly decrease the predictive accuracy of a traditional learning algorithm. This is because a learning algorithm aims to learn an optimal model for the query log by minimizing the expected risk:

$$\hat{\theta} = \arg \min_{\theta} \sum_{(q,y) \in D_q} P(D_q) \ell(q, y, \theta) \quad (6.1)$$

where D_q is query log data, q is a query, y is an *ODP* category, and $\ell(q, y, \theta)$ is a loss function with parameter θ .

However, since we have to assume that no labelled data is available from the query log, we have to learn a model from the Facebook data instead by minimizing the empirical risk:

$$\hat{\theta} = \arg \min_{\theta} \sum_{(l,y) \in D_f} P(D_f) \ell(l, y, \theta) \quad (6.2)$$

Note that here we have likes (l) instead of queries (q). If $P(D_q) = P(D_f)$, the two optimization problems are approximately equivalent. However, as we can observe from the comparison of the Facebook and search data (see Table 6.1), the two distributions $P(D_q)$ and $P(D_f)$ are different.

To predict demographic characteristics of the users in a query log, we essentially seek to obtain the conditional probability distribution $P(Y|Q, D_q)$, where Y denotes the demographic characteristic of a user who issued queries Q , and D_q denotes the query log. Note that $P(Y|Q, D_q) \neq P(Y|L, D_f)$ as discussed earlier.

Since we choose to represent each user by a probability distribution over ODP categories, $P(Y|Q, D_q)$ can be marginalized across ODP categories C :

$$P(Y|Q, D_q) = \sum_C P(Y|C, D_q) P(C|Q, D_q) \quad (6.3)$$

By Bayes' rule, $P(Y|C, D_q)$ is given by:

$$P(Y|C, D_q) = \frac{P(Y|D_q) P(C|Y, D_q)}{P(C|D_q)}, \quad (6.4)$$

where $P(Y|D_q)$ is the probability of class Y in the query log, which captures our prior knowledge about the relative frequencies of users of different demographics in a search engine. These quantities can be obtained from the search engine internal statistics, or publicly available statistics about the search users. On the other hand, $P(C|D_q)$ captures the relative frequencies of queries of category C . This quantity could be estimated from search logs, but can also be approximated from the ODP/DMOZ statistics assuming that the ODP corpus is statistically similar to the set of results returned by the search engine. $P(C|Y, D_q)$ is the probability

that a user with demographics Y is interested in category C when issuing a query. The key insight here is that we can assume that whether a user is interested in some category C or not depends on their demographics Y , independent of whether he or she is using Facebook or doing search. Therefore, it is reasonable to make the conditional independence assumption that

$$P(C|Y, D_q) = P(C|Y) = P(C|Y, D_f), \quad (6.5)$$

which means that $P(C|Y, D_q)$ can be estimated from Facebook data D_f . Let θ^Y denote the probability distribution $P(C|Y)$. In order to avoid problems of estimation due to sparsity of the data we estimate the parameter vector θ^Y using Bayesian Maximum A Posteriori (MAP) estimation. In particular, we estimate θ^Y by:

$$\hat{\theta}^Y = \arg \max_{\theta^Y} P(\theta^Y | D_f) = \arg \max_{\theta^Y} P(D_f | \theta^Y) P(\theta^Y). \quad (6.6)$$

This is a standard Bayesian estimation problem with a multinomial likelihood and a conjugate Dirichlet prior $P(\theta^Y)$ parameterized by pseudo-counts $\{\alpha_k\}$, ($\alpha = \sum_k \alpha_k$). If there is prior knowledge available, this can be taken into account, otherwise one can initialize the pseudo counts $\{\alpha_k\}$ uniformly. The resulting MAP solution is given by:

$$\theta_k^Y = \frac{N_k^Y + \alpha_k - 1}{N^Y + \alpha - K}, \quad (6.7)$$

where N_k^Y is the number of times the webpages, which are returned for the Likes of users of class Y , fall into the k th category, K is the total number of ODP categories, and N^Y is the total number of categories for webpages returned for the Likes of users of class Y . Note that we estimate the probability $P(C|Q, D_q)$ in Equation (6.3) in a similar way.

In summary, the methodology outlined above allows us to train a demographics classifier on users characterized by their collection of Facebook Likes, yet evaluate it on users characterized by their search query history. We believe that the two key ideas of a) creating a common representation in terms of ODP, and b) of mitigating

Table 6.1: The distribution of age and gender in search queries and Facebook Likes datasets.

Dataset	Teenage (10-18)	Youngster (19-24)	Young (25-34)	Mid-Age (35-49)	Elder (50+)	Male	Female
Social Dataset	3%	49%	32%	14%	2%	37%	63%
Search Dataset	2%	11%	24%	39%	24%	53%	47%

the data shift problem by breaking up the problem into separate estimation tasks for demographics given category and category given query history will be more generally applicable to problems in which labels are available, but are not directly linked with the representation of interest through suitable training data.

6.4 Data

myPersonality Dataset (Facebook) The myPersonality dataset was collected through the *myPersonality* Facebook application, which allowed its users to take real psychometric tests and receive feedback on their scores. In addition to the results of the tests, respondents could opt in to record their Facebook profile data to be used for the research purposes. The dataset contains detailed psychodemographic profiles of more than 6 million unique users from diverse age groups, backgrounds, and cultures. Respondents were motivated to answer honestly, as the only gratification they received for their participation was feedback on their results. We used a subset of myPersonality users from US described by their age, gender, political views, religion, and lists of their Facebook Likes. We filter out all Facebook Likes associated with less than ten users. The resulting dataset contains over 457,000 users, 122,000 unique Likes, and over 11 million associations between the users and Facebook Likes. Users’ religion and political views were stored as free text. Although the great majority of users simply have the

typical religion/party/philosophy names in those fields (e.g. Christian, Liberal), sometimes we had to use regular expression matching to extract the relevant information. For instance, “Christian - Baptist” was recoded as “Christian” and “I dont go to church because i wanna leave room in the pews for the sinners that need it -mr. magee” was ignored after mismatching all of our regular expressions.

Bing Query Logs (Search) We apply the models trained on the myPersonality dataset to infer the traits of users characterized by search queries. Search query logs were obtained from Bing and were collected between October 14, 2012 and October 28, 2012. We have selected queries submitted by the US users that were signed in with their Microsoft Live account while issuing their queries. In total, we have collected 133 million queries from 3.3 million unique users. Each user was also described by age and gender as reported in their Microsoft Live profiles.⁶

Differing distributions (Data Shift) Table 6.1 shows that the distributions of user demographics significantly differ between myPersonality and search query logs datasets. For instance, on average there are more young and female users in our Facebook data, which considering the nature of myPersonality test may not be surprising.⁷

6.5 Evaluation

For each user trait used here, we first train a model on 66% of Facebook users in myPersonality dataset and test it on the remaining 34%. We then apply the same

⁶In both samples only anonymous data was used. The user IDs were all anonymized such that the actual usernames could not be identified.

⁷It is important to note that the demographics reported here for our search and social datasets necessarily cannot be regarded as representative statistics for Bing and Facebook. The distributions in the datasets, particularly for the myPersonality data, are significantly affected by how the data is collected. The unique characteristics of the myPersonality test is likely to attract certain types of audience more than others. Readers are encouraged to refer to other sources (such as alexa.com) for more representative statistics.

model on search queries and repeat the classification for search users.

Evaluation on myPersonality Sample In the binary classification tasks such as predicting gender or political view (liberal vs. conservative) we use the area under the ROC curve (AUC) measure for evaluating accuracy. The ROC curves are created by plotting the ratio of true positive rate versus false positive rate at various threshold settings. We turn each of the multiclass classification tasks such as predicting religion (among Christian, Buddhist, Jew, Agnostic) into multiple binary classification problems (e.g. Buddhist or not Buddhist) and report the average values at the end.

Evaluation on Bing sample The age and gender information of the Bing users was obtained from their Microsoft Live profiles. Hence, we can repeat the same type of AUC evaluation, but this time with the labels coming from Microsoft Live accounts.

Religion and political views are not available in the Microsoft Live profiles, hence we do not have the ground-truth information on the individual user level. Therefore, we evaluate the accuracy of the trained classifiers on how well their output matches the officially reported state-level statistics. We first classify the religion and political views of individual users (e.g. religion = Christian) and aggregate those results on the state level (e.g. 74% Christians in California) by using users' location acquired from the IP address. We then look up the corresponding reported values for each state from publicly available official statistics (e.g., what percentage of Californians are Christian). Next, for each given class (e.g., Christianity) and each state, we calculate the percentage of search users that are classified in that category with respect to (1) our predictions and (2) official statistics. Finally, we compute the Pearson correlation value (ρ) between (1) and (2) and consider it as a proxy for the accuracy of the prediction.

Table 6.2: The Area under the ROC Curve (AUC) for different demographic prediction models. The numbers in the middle column show the AUC of a model trained on Facebook data for predicting the demographics of Facebook users. In the right column, the models trained based on Facebook data are tested on search query sample. The missing values “-” are used where the per-user ground-truth information is not available for AUC evaluation.

AUC	Facebook-Facebook	Facebook-Search
Gender	0.836	0.803
Age	0.771	0.735
Religion	0.758	-
Political view	0.739	-

6.6 Experiments

Using the compact ODP representation described earlier, we managed to model all users in both Facebook and search queries datasets. In comparison, an exact-match approach that compares the text of queries and Likes finds only 5.3% overlap by which only 36% of search users can be modelled and even for those there are often only few non-zero features.

Table 6.2 displays the evaluation results of the classifiers built on Facebook sample for inferring different demographics. The middle column (Facebook-Facebook) shows the AUC values when we trained and tested on the Facebook dataset. The right column (Facebook-Search) shows the Facebook model AUC on classifying Search users.

For gender classification, we train two separate classifiers; one for male, and one for female, each computing the probability of given gender based on the user profile (ODP features). Each user is compared against both of these classifiers, and the one producing the highest probability is used to set the class of gender.

The results in Table 6.2 show that the classification reaches 83% and 80% AUC respectively when tested on Facebook and Search samples. Not surprisingly, the AUC of a model trained based on the Facebook sample, is higher when it is tested on other users from the same dataset. However, the relative loss is not substantial, particularly considering the significant differences in the demographic distributions of these two sets (37% Male in Facebook dataset, compared to 53% among Search users).

For age classification, we grouped the users in each dataset into five separate age groups as listed in Table 6.1. For each age group we compute a model based on the training subset of users in our Facebook dataset. At testing, each user – in Facebook and Search datasets – is compared against these models, and the one producing the highest probability is used for classification. On the testing subset of Facebook users, the trained classifier achieves 77% AUC, while this number is slightly lower (73.5%) when applying the model on the Search sample.

To classify users’ religion, we first apply a set of regular expressions as described in Section 6.5 to assign the social users into four groups: {Christian, Jewish, Buddhist, and Agnostic/Unaffiliated }. These are also the four major religions in the United States according to U.S. Religious Landscape Survey,⁸ accounting respectively for of 78.4%, 1.7%, 0.7%, and 16.1% of the entire US population. We use these nationwide statistics as the prior when classifying the users in the Search dataset. The AUC while classifying users religion in the testing subset of Facebook dataset is 76%. Importantly, as there is no information about the religion on the Search user level, the accuracy of the classification was evaluated in terms of how well it predicted the state-level distributions as described in Section 6.5.

Figure 6.2 depicts the state distributions of Christians (top) and Buddhists (bottom) according to the U.S. Religious Landscape Survey on the left, and ac-

⁸<http://religions.pewforum.org>

ording to our predictions for the search engine users on the right.⁹ The models predict that depending on the state, 65.8%-95.3% of search users are Christians. These values are comparable to 69.1%-92.9% reported in the Landscape Survey. We also correctly identify the Mississippi state as the one with the highest ratio of Christians, and the states on the east coast with the lowest density ($\rho = 0.39$). Similarly the models predict 0.4%-5.7% of search users in the dataset to be Buddhist depending on the state, which is not far from the 0.5%-2.1% range reported in the Landscape Survey. The models predict Vermont, Oregon, California, and New Mexico to have the largest population of Buddhists, and apart from the former – that accounts for 0.001% of our dataset and hence is somewhat prone to noise – the remaining three are also listed as the top three Buddhist states in the Landscape Survey (Overall, $\rho = 0.53$).

Figure 6.3 demonstrates the spread of Agnostic (top) and Jewish (Bottom) people in the United States. The models predict 4.1%-27.6% of the search users in our dataset to be agnostic or unaffiliated with any particular religion. The official numbers from the Landscape Survey for this category lie closely between 6.1% and 28.3%. Consistent with the Landscape Survey, our models predict higher density of agnostics in North East and West, with the state of New Hampshire appearing on top of both – survey and predicted – lists ($\rho = 0.27$). According to our predictions based on search engine users, Jews account for 0.3%-5.0% of the US population depending on the state. These numbers are fairly consistent with the 0.5%-6.5% reported on the Landscape Survey ($\rho = 0.54$). We also correctly identify the states in the North East, in particular New York to have the highest density of Jewish people. This is yet again aligned with the Landscape Survey and historical documents about the Jewish settlements in the United States.¹⁰

We matched a set of regular expressions against the *Political view* field of users

⁹The states of Alaska and Hawaii do not appear on the Landscape Survey and hence are dropped from the analysis.

¹⁰http://en.wikipedia.org/wiki/American_Jews

in the Facebook dataset to group them into *liberal* (34%) and *conservative* (66%) categories. We ignored users that did not match any of the regular expressions in building our models. The distribution of liberal versus conservative in the social dataset is remarkably close to those reported by independent sources such as Gallup survey which reported 20.6% liberals versus 40% conservatives nationwide – the remainder of people in the poll were assigned to *moderate* and other groups.¹¹

As in previous experiments, we build the classifiers based on the ODP features of the users in the training subset (64%) of Facebook dataset. Applying the model on the remaining (34%) of users in that dataset produces the AUC of 0.74. We then apply the same model on the Search sample; the middle and bottom maps in Figure 6.5 illustrate the distribution of liberals and conservatives in the US. The middle map is generated based on the per-state statistics reported by the Gallup survey. The bottom map is generated by applying the classifier trained on the Facebook sample to Search users. The predicted class for each individual user contributes to generate the overall distribution for each of the states.

To enhance the visualization, the plots were produced with respect to the nationwide average so that the differences between states become more prominent. For instance, -0.10 would mean 10% more liberal, while 0.05 would suggest 5% more conservative than the nationwide average. The middle and bottom maps in Figure 6.5 reveal very similar distributions ($\rho= 0.72$). As expected, both maps look more blue on the East-West coasts, and more red in the so-called *Bible Belt* states. Oregon with an officially reported 13.8% swing towards liberals is the most noticeable mispredicted state; this was affected to some extent by the ambiguity of the queries related to *the civil war*, a college football rivalry in Oregon, which was particularly trendy during our sampling period.

It is commonly known that liberals are more likely to vote for the Democratic

¹¹Gallup poll, <http://bit.ly/hsceKj>

Table 6.3: The ODP categories with the highest information gain for different types of demographics.

Gender	Age	Religion	Political view
Sports/Basketball	Arts/Movies	Religion_and_Spirituality/Christianity	Politics/Liberalism
Games	Computers/Data_Communications	Religion_and_Spirituality/Religious_Studies	Politics/Conservatism
Sports/Soccer	Games	Religion_and_Spirituality/Scientology	Society/History
Shopping/Gifts	Shopping/Toys_and_Games	Society/History	Arts/Movies
Shopping/Jewellery	Computers/Software	News/Media	Science/Social_Sciences

Party and conservatives are more likely to vote for Republicans.¹² Thus, perhaps it is not entirely surprising to find similarities in how the states were split between Democrats and Republicans in the recent 2012 US presidential election (top map in Figure 6.5).

6.7 Importance of ODP categories

In this section we show the importance of each category in predicting a given type according to its *information gain* computed in a leave-one-out fashion. That is, for each ODP category C (e.g. Arts/Movies), and a given demographic type Y (e.g. Gender), we first calculate the prior values according to all other 219 categories in our data, and then calculate the change in information entropy when C is considered as,

$$IG(Y, C) = H(Y) - H(Y|C) \quad (6.8)$$

Here, $H(Y)$ represents the prior entropy for the demographic type Y across all users, and $H(Y|C)$ is the same value conditioned on observing category C in the user's profile. Table 6.3 shows the categories with the highest information gains for classifying each of the demographics. For classifying gender, sport and shopping related categories are most effective. Art/Movies, Games, Shopping/-Toys_and_Games and computer-related categories are best in discriminating between different age groups. For religion, subcategories of Religion_and_Spirituality

¹²Gallup Politics, <http://bit.ly/AoyIg4>, and Rasmussen Report, <http://bit.ly/L85SmV>

are the most important features, and for politics – not surprisingly – Politics/Liberalism and Politics/Conservatism have the highest information gain values for distinguishing between liberals and conservatives.

We also calculate the *influence* (β) of each category $c \in C$ (e.g. Arts/Movies) in classifying a given demographic type to a particular class $y \in Y$ (e.g. Gender = Male) by,

$$\beta = \frac{P(C = c|Y = y) - \mu}{\mu} \quad (6.9)$$

where μ represents the average probability of class $c \in C$, for all values of $y \in Y$. That is,

$$\mu = \frac{\sum_{y \in Y} P(c|y)}{|Y|} \quad (6.10)$$

When ranking categories by Equation (6.9), for gender, we found Shopping/{Jewelry, Health, Pets, Craft}, Arts/Design, and Society/Relationship as the most *influential* categories for classifying females. For males, Shopping/Gift, Sports sub-categories, Games and Recreation/Guns had the highest influence. For political views, Politics/Conservatism, and Society/{Military, Politics, Religion_and_Spirituality} had the highest β scores for conservatives, while for liberals Society/Gay, Lesbian, and Bisexual, Politics/Liberalism, and Computers/Artificial_Intelligence were ranked highest.

For age, Kids_and_Teens/Health had the highest β among teenagers. Adult/Society and Sports/Wrestling were the highest-ranked categories for youngs and youngsters. Shopping/Jewellery, and Business/Hospitality were closely ranked on top for mid-age users, while Shopping/Ethnic_and_Regional and News/Media were the top two for elders.

Finally for religion, Religion_and_Spirituality/Christianity, and Religion_and_Spirituality/Scientology had respectively the highest bias towards Christians and Buddhists. For Jews, somewhat surprisingly Computers/Computer_Science was ranked highest, while agnostics had the strongest negative biases towards Religion_and_Spirituality/Religious_Studies, and Religion_and_Spirituality/Scientology.

6.8 Conclusion

In this work, we addressed the problem of inferring users traits – namely age, gender, religion and political view – from their search queries. We trained our predictive models on a sample of Facebook users that had agreed to provide their Likes and other profile information for research purposes. To the best of our knowledge, this is the first study that infers the demographics of search users based on the models trained on the independent social datasets.

We demonstrated that both Facebook Likes and search queries can be translated into a common representation via mapping to ODP categories. In addition, we addressed the data-shift problem by breaking up the problem into separate estimation tasks for demographics given category, and category given query history.

Our experimental results on a large scale query log of a commercial search engine confirms that the demographics of search users can be accurately predicted based on models trained on an independent social data. The trained classifiers achieved 80% and 74% AUC respectively for classifying gender and age. For various religious and political views the models consistently ranked the US states close to their rankings reported in the official statistics (Pearson $\rho > 0.72$ in all our experiments).

For future work, we are interested in expanding the models to capture other types of user traits, such as personality, intelligence, happiness, or interests and measuring the applications of those inferred traits in personalization, reranking and monetization of the search results.

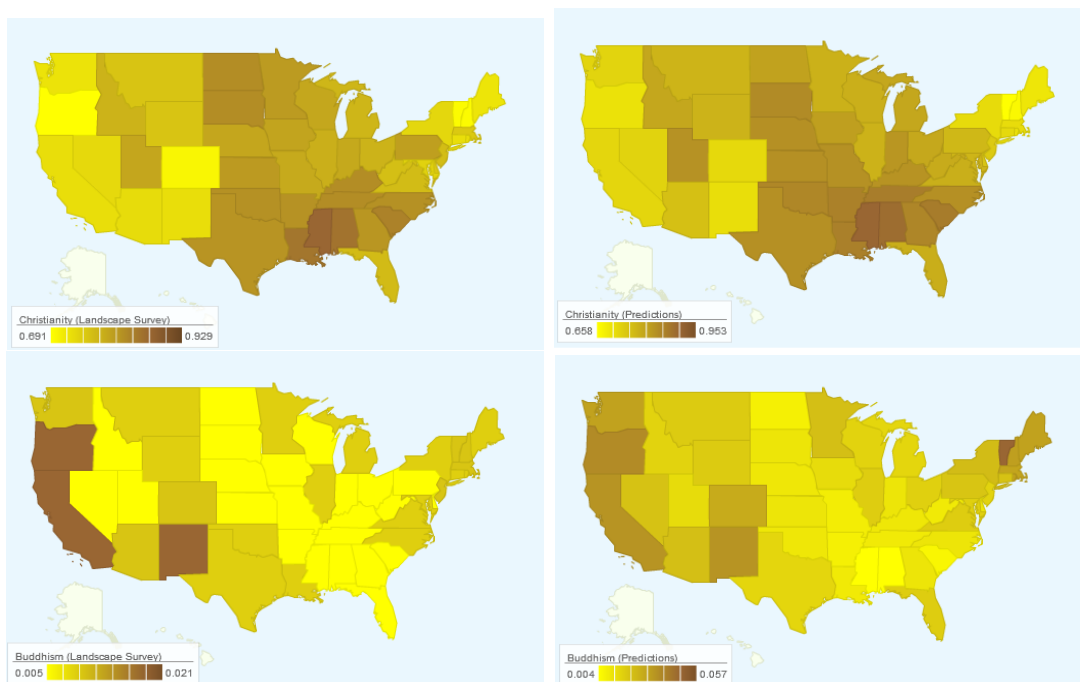


Figure 6.2: **(Top-Left)** The distribution of Christians in the *Contiguous* United States according to the U.S. Religious Landscape Survey. **(Top-Right)** The distribution of Christians in the US as predicted based on user queries. The Pearson correlation (ρ) is 0.39. **(Bottom-Left)** The distribution of Buddhism in the *Contiguous* United States according to the U.S. Religious Landscape Survey. **(Bottom-Right)** The distribution of Buddhism in the US as predicted based on user queries. The Pearson correlation (ρ) is 0.53. The spectrum bar at the left corner of each map specifies the scale and the corresponding color codes.

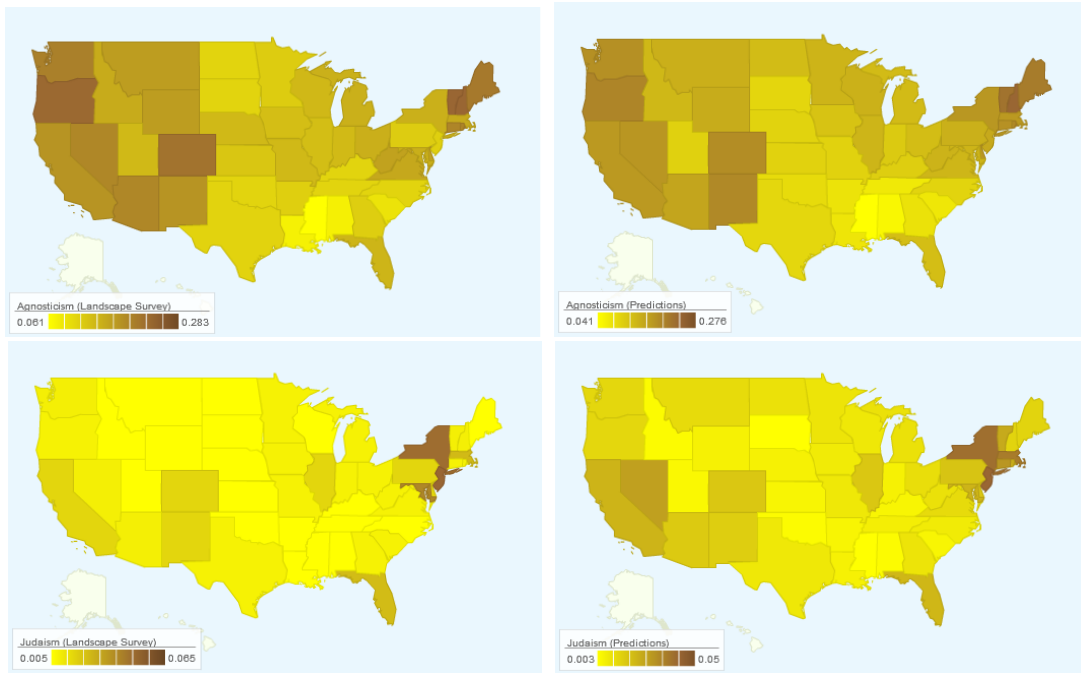


Figure 6.3: **(Top-Left)** The distribution of Agnostics in the *Contiguous* United States according to the U.S. Religious Landscape Survey. **(Top-Right)** The distribution of Agnostics in the US as predicted based on user queries. The Pearson correlation (ρ) is 0.27. **(Bottom-Left)** The distribution of Judaism in the *Contiguous* United States according to the U.S. Religious Landscape Survey. **(Bottom-Right)** The distribution of Judaism in the US as predicted based on user queries. The Pearson correlation (ρ) is 0.54. The spectrum bar at the left corner of each map specifies the scale and the corresponding color codes.

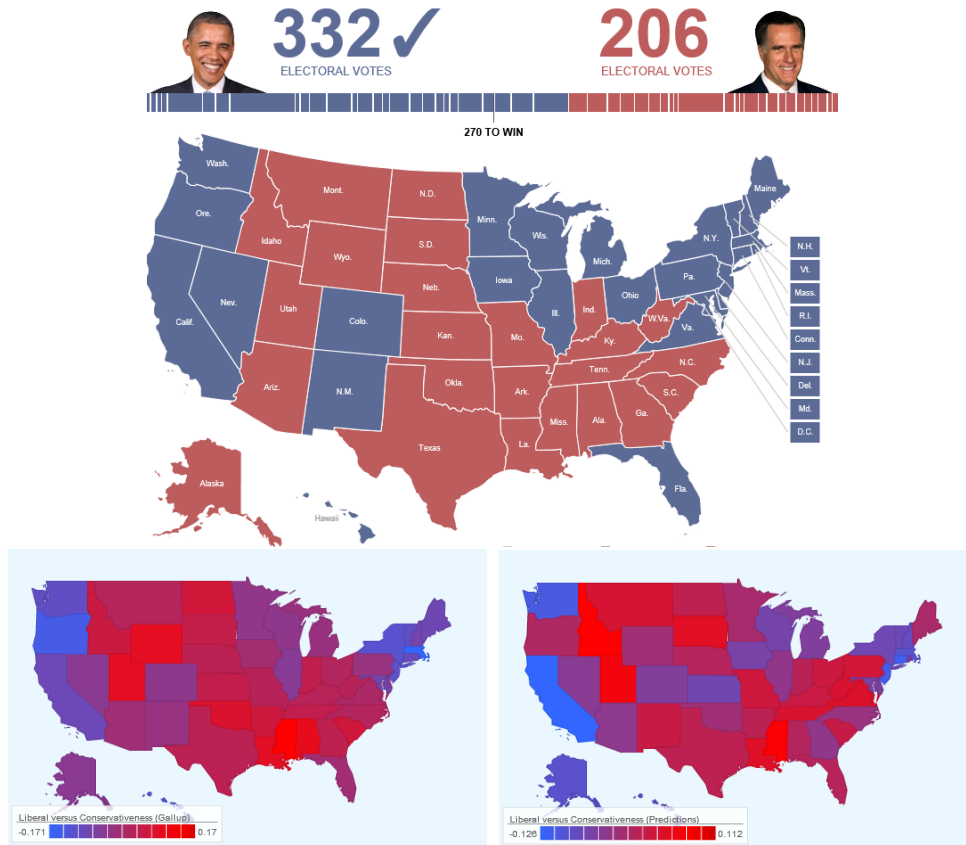


Figure 6.4: **(Top)** The outcome of 2012 the US presidential election according to The Huffington Post. The blue states were won by Democrats and the red states by Republicans. **(Bottom-Left)** The distribution of conservatives versus liberals according to an independent poll – Gallup. **(Bottom-Right)** Liberal vs. conservative predictions on Bing users based on the models learned according to Facebook data. The Pearson correlation (ρ) between the Gallup data and our per-state predictions is 0.72. The spectrum bar at the left corner of each map specifies the scale and the corresponding color codes.

CHAPTER 7

Learning to Recommend Related Entities to Search Users

7.1 Introduction

Traditionally, web search engines have led users toward web pages chosen by lexical matches against the search string. However with the introduction of *knowledge bases* over the past few years, commercial search engines are moving towards retrieval based on a semantic understanding of the user query. The knowledge base is being used to provide popular facts about people, places, and things alongside traditional search results. It allows search to evolve from returning pages that match query terms to finding *entities* that the words describe.

A knowledge base is a centralized repository of content about entities, their attributes and mutual relationships. Well-known examples of knowledge bases include Freebase, YAGO, Microsoft Satori, and Google Knowledge Graph. For instance, Freebase consists of a large set of metadata about movies, music, books, well-known people, and things. A subset of the entities and their relations in Freebase is depicted in Figure 7.1. It includes entities corresponding to four people, two movies and their genres. The links between the entities represent their relationships, such as “*Adam McKay* is the director of *Anchorman*” and “*Kristen Wiig* is an actor appearing in *Anchorman 2*”. In this example, “director”, “actor” and “genre” are attributes of the entity *Anchorman*.

With the introduction of a knowledge base, a web search engine enables users

to search for things – movies, celebrities, landmarks and more – and instantly get rich information relevant to the queries. Figure 7.2 shows an example search result for the query “pacific rim” together with its entity pane provided by a commercial search engine. The search engine recognizes that “pacific rim” is the title of a movie corresponding to an entity in the knowledge base. We refer to the entity a user searches for as the *main entity*. An entity pane that presents information about the main entity shows up to the right of the regular search results. On the entity pane, in addition to the description of the movie *Pacific Rim*, a list of movies related to *Pacific Rim* is also visually presented below. We refer to an entity related to the search as a *related entity*. The provided related entities allow users to quickly access other relevant entities and offer the ability to explore more information within the same search session. In order to keep users engaged, it is important to develop a recommendation model that generates related entities closely matched with their interests.

Currently, major search engines recommend related entities based on their similarities to the main entity that the user searched for. There are various measures of the similarity between a main entity and an entity to recommend. A common measure is the frequency of the two entities being co-clicked in the same session across all search users. A related entity is recommended if and only if it is frequently co-clicked with the main entity. This co-click based approach essentially maximizes the likelihood that people agree on the relatedness irrespective of any individual user. Such a global recommendation method brings the same list of related entities to every user who searches for the same main entity, as user-specific information is completely ignored. But the same recommendation cannot satisfy users with distinct interests. For example, given a movie as the main entity, one user may be interested in viewing the other movie entities with the same director, while another user may want to view the movie entities from the same genre.

To the best of our knowledge, no work has been done on developing a rec-

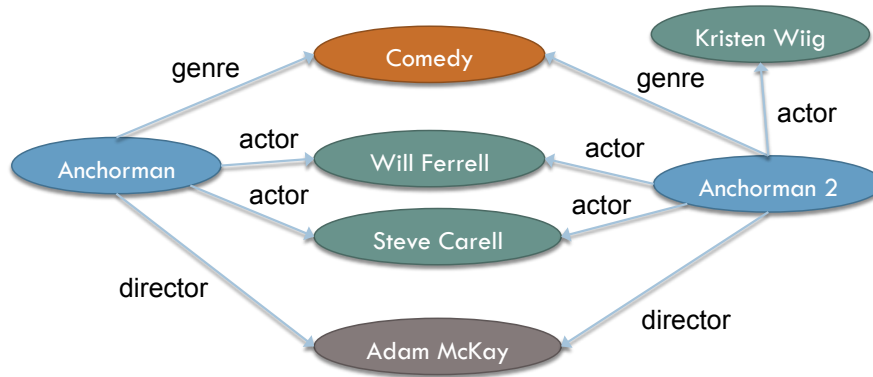


Figure 7.1: Example of the entities and their relations taken from Freebase

ommendation model for a search engine to tailor related entities to an individual user’s unique taste and preference. To personalize recommendations, we need to build user-specific profiles from their interactions with the search engine. In this work, the users’ interactions are collected in the *search click log* and the *entity pane log*. The search click log stores history of user clicks on URLs, while the entity pane log stores clicks on the entity pane. In this work, we aim to build a probabilistic recommendation model that can customize the suggested entities, which are related to a given main entity, based on the user’s past history stored in the usage logs.

Despite considerable research on the *search click log* over the last decade, little is known about the emerging *entity pane log*. This work also represents the first study exploiting the entity pane’s implicit user feedback for entity recommendation. Our empirical studies find that the entity pane click-through rates (CTR) play important roles in enhancing recommendation quality of related entities. Therefore, we include these strong CTR signals in the recommendation model.

In addition to CTRs, our recommendation model involves three important dimensions: *user*, *main entity*, and *related entity*. Without the user dimension, the model would degenerate to a global recommendation method which fails to

The image shows a search engine results page for the query "pacific rim". The search bar at the top contains "pacific rim" and a magnifying glass icon. Below the search bar, the results are organized into several sections:

- Search Summary:** "17,400,000 RESULTS" with filters for "Any time" and "Near Redmond, Washington".
- Warner Bros. Pictures and Legendary Pictures Pacific Rim**: A link to the official website with a brief description of the film.
- Landmark Crest Cinema Centre**: A list of showtimes for various locations.
- Pacific Rim (2013) - IMDb**: A link to the IMDb page with a rating of 7.6/10 and a brief plot summary.
- Videos of pacific rim**: A section with video thumbnails and links to official trailers and content on YouTube.
- Pacific Rim (film) - Wikipedia, the free encyclopedia**: A link to the Wikipedia page with a brief plot summary.

On the right side of the page, there is a detailed "Main Entity" pane for "Pacific Rim (2013)". This pane includes:

- Entity Name:** Pacific Rim (2013)
- Image:** A movie poster for Pacific Rim.
- Description:** "When legions of monstrous creatures, known as Kaiju, started rising from the sea, a war began that would take millions of lives and consume humanity's resources for years on end. To combat the giant Kaiju, a special type of weapon was devised: massive robots, called Jaegers, ..."
- Watch trailer:** A link to watch the trailer on MSN.
- Summary:** PG-13 · 2hr 11min · Action/Adventure
- Director:** Guillermo del Toro
- Reviews:** A star rating of 7.6/10, with 64% positive reviews on Metacritic and 81% positive reviews on Flixster.

Below the "Main Entity" pane is a "Related Entities" section, which features a row of four movie posters with their titles: "Man of Steel", "The Wolverine", "The Lone Ranger", and "World War Z".

Figure 7.2: Example of search results with the entity pane taken from a commercial web search engine

personalize suggested entities, as discussed above. On the other hand, if recommendations were based purely on the user dimension, while totally ignoring main entities, then the suggested entities would be utterly unrelated to the searches. The interactive feedback in the usage logs reveals the three-way correlations among these three dimensions. The recommendation model aims to discover and exploit their ternary relationships. We refer to our probabilistic recommendation model as Three-way Entity Model, abbreviated as TEM.

To determine the parameters in TEM, we propose learning their optimal values from a training set of observations constructed from the entity pane log. As mentioned above, this log contains user feedback on the relatedness of recommended entities. Positive observations can be readily derived from click feedback by interpreting a user click as a vote in favor of relatedness. Nevertheless, it is nontrivial to derive negative observations, since a non-click may not indicate the

absence of relatedness. We propose a principled solution to this issue, specialized for the problem of ranking related entities.

The major contributions of our work are summarized as follows:

1. This work provides the first solution – the probabilistic model TEM– for a search engine to personalize its recommendation of related entities. The recommended entities are customized to be not only related to the given main entity, but also tailored to the user’s interest and preference.
2. The TEM model leverages three data sources: *knowledge base*, *search click log*, and *entity pane log*. This is the first work to utilize the entity pane log to recommend related entities. Specifically, the CTRs derived from the entity pane log turn out to be strong signals for entity recommendation.
3. The TEM model uncovers the underlying three-way relationships among *user*, *main entity*, and *related entity*. Jointly modeling all three dimensions prevents TEM from making static or irrelevant recommendations. An inference technique is introduced to learn the parameters of TEM.
4. We propose a principled method for training set construction to work around the problem of missing negative samples. The proposed method is specifically designed for ranking related entities.
5. We conducted extensive experiments with two real-world datasets of different domains collected from a commercial web search engine. The experimental results demonstrate that TEM with our probabilistic framework significantly outperforms the state of the art used by a commercial search engine. It confirms the effectiveness of TEM and our probabilistic framework in entity recommendation and the efficacy of personalization.

7.2 Related Work

A research topic related to our work is personalized web services, including web search and entity/news recommendation, although the tasks are quite different. Micarelli et al. [78] provided a summary of research works on this topic. Personalized search exploits user search histories to deliver more relevant results than those provided by traditional search engines [98]. Unlike the task addressed in this work, Blanco et al. [22] worked on a fundamentally different entity recommendation task. The goal of their work was to recommend possible future queries related to the user’s current search query based on a knowledge base. In their paper, the future queries were referred as to related entities, as opposed to the related entities in our context. Chu and Park [34] proposed a feature-based bilinear regression framework for personalized recommendation on news content. This approach greatly alleviated the cold-start issue of recommending for new users by leveraging interest patterns in user profiles recognized from regression over historical interactive feedback. Sun et al. [99] introduced CubeSVD to perform three-way data analysis for personalized search. Similar to personalized search, our work exploits prior user actions to model their interests for personalized recommendation on related entities.

Over recent decades, a few studies have been conducted on three-way data analysis. Acar and Yener [1] gave an overview of multiway models, algorithms as well as their applications in diverse disciplines. These studies commonly represented observational data as a third-order tensor, which is a higher-order generalization of a vector and a matrix. A three-way model was then constructed for extracting hidden structures and capturing underlying correlations between variables in the third-order tensor. A well-known three-way model, called Tucker3, was introduced by Tucker [103, 35]. It is an extension of singular value decomposition to third-order tensors. Tucker3 has been successful in many applications

	User ID	Time	Main entity	Related entity	Rank	Click
Instance	32	7/9/2013 10:32:26	Pacific Rim	Man of Steel	1	0
	32	7/9/2013 10:32:26	Pacific Rim	The Wolverine	2	0
	32	7/9/2013 10:32:26	Pacific Rim	The Lone Ranger	3	1
	32	7/9/2013 10:32:26	Pacific Rim	World War Z	4	0
Page impression 1						
Page impression 2	498	6/16/2013 15:16:41	Leonardo DiCaprio	Kate Winslet	1	0
	498	6/16/2013 15:16:41	Leonardo DiCaprio	Baz Luhrmann	2	0
	498	6/16/2013 15:16:41	Leonardo DiCaprio	Johnny Depp	3	1

Figure 7.3: Sample records taken from an entity pane log

[99, 105].

Three-way data analysis has been widely performed in the context of multi-verse recommendation. Karatzoglou et al. [66] introduced a collaborative filtering method based on third-order tensor decomposition to provide context-aware recommendations. Rendle and Schmidt-Thieme [92] used the tensor decomposition technique in recommending to users tags for annotating specific items in social tagging systems. Despite its success in recommender systems, tensor decomposition does not apply to related entity recommendation. In particular, tensor decomposition suffers from the cold-start problem, as it represents each object in the system with a unique ID. Given the knowledge base and the usage logs, tensor decomposition cannot utilize the valuable information derived from the various nature of data sources. In order to do so, we developed TEM, a new probabilistic model for three-way data analysis.

7.3 Problem Statement

In a nutshell, the objective of this work is to recommend to the user a ranked list of entities relevant to the main entity by leveraging three pieces of information: *knowledge base*, *search click log*, and *entity pane log*.

Figure 7.3 presents a few sample records from the entity pane click log of a real search engine. Each row represents an *instance* indicating whether user u clicked related entity r given main entity m . There are two *page impressions* in the table, each of which indicates the list of related entities recommended for a given (u, m) pair. The *Rank* column gives the rank of each related entity in recommendation lists, and the *Click* column indicates whether related entities were clicked or not (1 for click, 0 for no click).

For notational convenience, let U denote the total number of unique users in the log, M denote the total number of main entities, and R denote the total number of related entities. The notations used throughout this chapter are given in Table 7.1. Some of the notations will be explained in later sections.

As discussed above, a click event is naturally associated with three salient dimensions: $User \times Main\ entity \times Related\ entity$.

[User dimension]

The user dimension targets user interest patterns, building search profiles by logging user interactions with the search engine. In this work, a user profile maps a user to a vector of entities and attributes representing the user’s interests. In order to model user interest as accurately as possible, we collect click history from two sources: *search click log* and *entity pane log*. The former records user click history on URLs, while the latter reflects user click history on entities.

Since the *search click log* reports user clicks on URLs, but we are looking for user interest in entities, we need a mapping from URLs to entities. Fortunately, with the help of the open source Freebase¹ knowledge base, it is easy to map users to the entities they are interested in. An illustration is shown in Figure 7.4.

Each entity in Freebase is linked to some URLs that are related to this entity. For example, for the movie *Avatar*², by utilizing the relationships “/com-

¹<http://www.freebase.com/>

²<http://www.freebase.com/m/0bth54>

Table 7.1: Notations used throughout this chapter

Notation	Description
X	Feature matrix for users
Y	Feature matrix for main entities
Z	Feature matrix for related entities
u	User identity
m	Main entity
r	Related entity
\mathbf{x}_u	Feature vector for user u
\mathbf{y}_m	Feature vector for main entity m
\mathbf{z}_r	Feature vector for related entity r
U	Number of unique users
M	Number of main entities
R	Number of related entities
I	Number of features for each user
J	Number of features for each main entity
K	Number of features for each related entity
Θ	Model parameter
η, β	Weight coefficients
o	Preference relation

mon/topic/official_website” and “/common/topic/topic_equivalent_webpage”, we can obtain this movie’s official site, IMDb pages as well as Wikipedia pages, etc. Moreover, other information related to this movie is available to us, including actors, directors, genres, producers, etc. With the help of this data, user-clicked URLs in *search click log* can be mapped to corresponding entities in Freebase, as demonstrated in Figure 7.4. The attributes of these entities can also be obtained from the Freebase knowledge base.

For the *entity pane log*, as shown in Figure 7.3, the clicked entities are already known, so we can simply extract corresponding attributes from Freebase to represent user interests. By combining the above signals, entities and their attributes

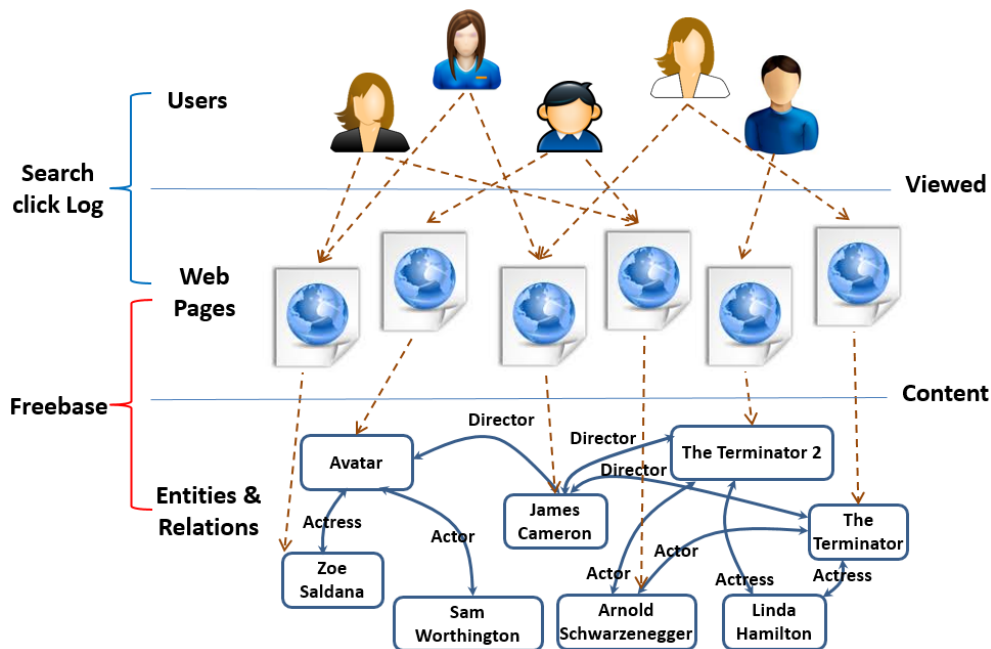


Figure 7.4: Illustration on joining search click log with Freebase knowledge base can be used to model users in a naturally way. For instance, a group of users may often view entities about action movies, while another group may prefer basketball players. If users are modeled appropriately through their usage patterns, these preferences and interests should help the search engine recommend related entities more accurately. Some sample features we extract for users are shown in Table 7.3 in Section 7.5.

Formally, each user is represented as a vector of features, denoted by \mathbf{x} , where $\mathbf{x} \in \mathbb{R}^I$ and I is the number of user features.

[Main entity]

In nature, the main entity reflects the search user's current search interest. In addition to user profiling by modeling his or her interest pattern based on the usage log, it is important to capture a user's current search intent expressed by the main entity, which provides valuable context. Ignoring main entities will compromise the performance of a recommendation model. In particular, if related entities are

obtained based purely on the user’s past preferences, while neglecting to model his or her current interest, then the recommended entities will be completely independent of what the user is searching for, leading to dissatisfaction.

The feature space for main entities is spanned by their attributes extracted from the knowledge base. Each main entity is represented as a vector of features, denoted by \mathbf{y} , where $\mathbf{y} \in \mathbb{R}^J$ and J is the dimensionality of the feature space for main entities.

[Related entity]

A user may click a related entity when it is aligned with both the user’s interest pattern and current intent. User clicks on related entities are the interactive feedback used to relate patterns in user features to main entities. For example, suppose there is a fan of the film director *Steven Spielberg*. (Such a user can be identified from his or her past usage pattern.) Given a movie as a main entity, a good recommender should recommend the other movies related not just to the given entity, but also directed by *Steven Spielberg*, instead of recommending movies related in other ways, such as sharing the same actors.

Each related entity is represented as a column vector of features, denoted by \mathbf{z} , where $\mathbf{z} \in \mathbb{R}^K$ and K is the number of features for each related entity.

As we have argued, all three dimensions, *User* \times *Main entity* \times *Related entity*, can improve entity recommendation, which motivates our joint modeling of these factors. The joint model is intended to capture structural dependencies of the three dimensions, revealing the underlying ternary relations.

Problem Statement. *Given the feature representations for users \mathbf{x} , main entities \mathbf{y} and related entities \mathbf{z} , we aim to develop a recommendation model that uncovers the three-way correlations among them to recommend a ranked list of entities related to a given main entity for any user.*

7.4 Three-way Entity Model

In this section, we present a three-way probabilistic model, TEM, designed to uncover the pattern correlations among \mathbf{x} , \mathbf{y} and \mathbf{z} for recommending related entities. More specifically, we first define a real-valued function $\Psi_{umr}(\Theta)$ of the model parameter Θ which captures the ternary relationship among the three dimensions. A likelihood function is then employed to relate the values of $\Psi_{umr}(\Theta)$ to observed actions on related entities. Finally, the parameter Θ is obtained by performing inference on TEM. The effect of the three-way interactions will be analyzed in this section.

7.4.1 Trilinear function

To jointly model users, main entities, and related entities, we define a trilinear function Φ_{umr} of \mathbf{x}_u , \mathbf{y}_m , and \mathbf{z}_r as follows:

$$\Phi_{umr}(\eta) = \sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K \eta_{ijk} \cdot x_{ui} \cdot y_{mj} \cdot z_{rk}, \quad (7.1)$$

where \mathbf{x}_u denotes the feature vector for user u , \mathbf{y}_m denotes the feature vector for main entity m , and \mathbf{z}_r denotes the feature vector for related entity r . x_{ui} is the i -th feature of \mathbf{x}_u , y_{mj} is the j -th feature of \mathbf{y}_m , and z_{rk} is the k -th feature of \mathbf{z}_r . η consists of a set of weight coefficients, which is introduced to capture the associations among the three objects \mathbf{x}_u , \mathbf{y}_m , and \mathbf{z}_r . The weight η_{ijk} quantifies the affinity of three features x_{ui} , y_{mj} , and z_{rk} . Note that η can be represented as a third-order tensor, where the value of each entry η_{ijk} will be learned from historical logs.

In order for the trilinear function to capture the pairwise associations between the three dimensions, we prepend a 1 at the beginning of each feature vector. As

a result, the users, main entities, and related entities are represented as:

$$\begin{aligned}\mathbf{x}_u &= [1, x_{u1}, x_{u2}, \dots, x_{uI}]^T, \\ \mathbf{y}_m &= [1, y_{m1}, y_{m2}, \dots, y_{mJ}]^T, \\ \mathbf{z}_r &= [1, z_{r1}, z_{r2}, \dots, z_{rK}]^T.\end{aligned}$$

Notice that when there is a large number of features for each dimension, $\eta \in \mathbb{R}^{(I+1) \times (J+1) \times (K+1)}$ becomes a huge tensor for which inference is intractable. To overcome this problem, we need to reduce the dimensionality of each feature vector. Given massive training data, we resort to random projections [60] for dimensionality reduction. Random projections essentially project each feature space onto a random lower-dimensional subspace, which yield results comparable to conventional dimensionality reduction approaches such as Principal Component Analysis (PCA). However, random projections are significantly less computationally expensive than PCA. We study the effect of random projections on entity recommendation in the experiment section.

7.4.2 CTR incorporation

The three-way associations, which are systematically modeled by the trilinear function $\Phi_{umr}(\eta)$, contribute an important indicator to entity recommendation, especially for the rare/new entities for which we have zero or insufficient click data. To further enhance the recommendation quality of popular entities, we derive CTR features from the interactive feedback collected in the entity pane log. The CTRs have been shown to be strong signals for various recommendation tasks [77, 59]. CTR is defined as the ratio of the number of clicks on a certain related entity and the number of page impressions in which the related entity is presented. We extract three sets of CTRs from the entity pane log:

1. $CTR(r)$: CTRs on related entities

2. $CTR(m, r)$: CTRs on main entities and related entities

3. $CTR(u, m, r)$: CTRs on users, main entities, and related entities

Following hybrid approaches proposed for personalized search and recommendation [27, 4, 34], we integrate the trilinear function $\Phi_{umr}(\eta)$ with the CTR features, and define a real-valued function $\Psi_{umr}(\Theta)$ as:

$$\begin{aligned}\Psi_{umr}(\Theta) &= \Phi_{umr}(\eta) + \beta^T \mathbf{c}^{umr} \\ &= \sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K \eta_{ijk} \cdot x_{ui} \cdot y_{mj} \cdot z_{rk} + \beta^T \mathbf{c}^{umr},\end{aligned}\quad (7.2)$$

where \mathbf{c}^{umr} is a vector of CTR features specific to user u , main entity m and related entity r . β is a vector of weight coefficients. $\Theta = (\eta, \beta)$ consists of all the parameters to be learned from historical logs.

7.4.3 Likelihood function

In this subsection, we introduce a likelihood function to relate the values of $\Psi_{umr}(\Theta)$ to the click log collected from the entity pane. The click log provides user preferences for related entities by keeping track of clicks as implicit feedback. One important fact about the click log is that only positive observations are available - each click can be considered as positive feedback for the corresponding triple (u, m, r) indicating that user u is interested in viewing entity r , which is related to main entity m . However, the non-clicked triples (u, m, r) (i.e., given main entity m , user u did not click recommended entity r on the entity pane), do not provide such clear conclusions. There are at least two different interpretations for any non-clicked triple. One possibility is negative feedback, meaning that the user was not interested in the recommended entity. Another possibility is that the user did not even see the entity, in which case the user's interested in the entity is unknown.

If we simply ignore all non-clicked triples, typical machine learning algorithms

are not able to learn anything from the positive observations alone. One may opt to consider the non-clicked triples as negative feedback. More specifically, training data is created by assigning positive class labels to clicked triples, and negative class labels to non-clicked triples. The problem with this approach is that all non-clicked triples the algorithm predicts in the future are presented to the learning algorithm as negative observations. This approach misinterprets non-clicked triples, which are actually missing values.

To address this problem, we use triple pairs as training data instead of individual triples. As opposed to replacing non-clicked triples with negative observations, we assume that users prefer the related entities they clicked over all other non-clicked ones on the same page impression. More specifically, given two triples (u, m, r_i) and (u, m, r_j) in the same page impression, user u prefers entity r_i over entity r_j if and only if r_i was clicked by u while r_j was not, which is denoted by $r_i^{u,m} \succ r_j^{u,m}$. Note that this assumption reasonably disregards click position bias, given the fact that only several related entities are presented in each page impression. This is different from the long lists of web search results, in which users are prone to click top ranked pages.

We create training data \mathcal{D} by including all preference relations induced, as follows:

$$\mathcal{D} = \{(u, m, r_i, r_j) | r_i^{u,m} \succ r_j^{u,m} \vee r_j^{u,m} \succ r_i^{u,m}\}, \quad (7.3)$$

where each preference relation $o = (u, m, r_i, r_j)$ is considered as a training sample. For the entities that are both clicked by a user, we cannot infer any preference. The same is true for two entities either of which a user did not click. The running example in Figure 7.5 shows the preference relations induced by the click feedback in the entity pane log. In the first page impression, as the user clicked the related entity *The Lone Ranger*, we infer that he or she prefers *The Lone Ranger* over the other three recommended movies, indicated by the arrows in the figure. Similarly, it can be inferred that, for the second page impression, the user is more interested

User ID	Time	Main entity	Related entity	Rank	Click
32	7/9/2013 10:32:26	Pacific Rim	Man of Steel	1	0
32	7/9/2013 10:32:26	Pacific Rim	The Wolverine	2	0
32	7/9/2013 10:32:26	Pacific Rim	The Lone Ranger	3	1
32	7/9/2013 10:32:26	Pacific Rim	World War Z	4	0
498	6/16/2013 15:16:41	Leonardo DiCaprio	Kate Winslet	1	0
498	6/16/2013 15:16:41	Leonardo DiCaprio	Baz Luhrmann	2	0
498	6/16/2013 15:16:41	Leonardo DiCaprio	Johnny Depp	3	1

Figure 7.5: Preference relations induced by the click feedback in the entity pane log

in *Johnny Depp* than the others.

A logistic function $\mathcal{F}(\cdot)$ as the likelihood function is then employed to relate the values of $\Psi_{umr}(\Theta)$ to the pairwise preference, as follows:

$$\begin{aligned}
& p(r_i^{u,m} \succ r_j^{u,m} | \Psi_{umr_i}(\Theta), \Psi_{umr_j}(\Theta)) \\
&= \frac{1}{1 + e^{-g_{r_i,r_j}^{u,m}(\Psi_{umr_i}(\Theta) - \Psi_{umr_j}(\Theta))}} \\
&= \mathcal{F}(g_{r_i,r_j}^{u,m}(\Psi_{umr_i}(\Theta) - \Psi_{umr_j}(\Theta))), \tag{7.4}
\end{aligned}$$

where $g_{r_i,r_j}^{u,m} \in \{-1, 1\}$ denotes whether user u clicks r_i or r_j for main entity m :

$$g_{r_i,r_j}^{u,m} = \begin{cases} 1 & \text{if } u \text{ clicks } r_i \text{ given } m, \\ -1 & \text{if } u \text{ clicks } r_j \text{ given } m. \end{cases}$$

The probability $p(r_i^{u,m} \succ r_j^{u,m} | \Psi_{umr_i}(\Theta), \Psi_{umr_j}(\Theta))$ gives the likelihood that user u prefers entity r_i over entity r_j , both related to main entity m . Given the inferred parameter Θ , the likelihood of observing all preference relations in training data

is then given by:

$$\begin{aligned}
p(\mathcal{D}|\Theta) &= \prod_{(u,m,r_i,r_j)\in\mathcal{D}} p(r_i^{u,m} \succ r_j^{u,m} | \Psi_{umr_i}(\Theta), \Psi_{umr_j}(\Theta)) \\
&= \prod_{(u,m,r_i,r_j)\in\mathcal{D}} \frac{1}{1 + e^{-g_{r_i,r_j}^{u,m}(\Psi_{umr_i}(\Theta) - \Psi_{umr_j}(\Theta))}} \\
&= \prod_{(u,m,r_i,r_j)\in\mathcal{D}} \mathcal{F}(g_{r_i,r_j}^{u,m}(\Psi_{umr_i}(\Theta) - \Psi_{umr_j}(\Theta))). \tag{7.5}
\end{aligned}$$

7.4.4 TEM & Inference

As discussed above, we need to learn the parameter Θ (i.e., η and β) from observed preference relations induced by user clicks on the entity pane, so that related entities can be recommended in the future.

For notational clarity, we define $\bar{\Theta}$ as a vector concatenating all the entries in η and β . The $\bar{\Theta}$ is considered as a random variable, and assumed to follow a Gaussian distribution:

$$\bar{\Theta} \sim \text{Gaussian}(\mu, \Sigma). \tag{7.6}$$

We impose a zero-mean isotropic Gaussian prior on the variable $\bar{\Theta}$, i.e.,

$$p(\bar{\Theta}) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{\sum_i \bar{\theta}_i^2}{2\sigma^2}}. \tag{7.7}$$

The graphical representation of the probabilistic model TEM is given in Figure 7.6. First, η is sampled from a Gaussian distribution. Given the η as well as features \mathbf{x}_u , \mathbf{y}_m , and \mathbf{z}_r , by Equation (7.1) we obtain the value of function $\Phi_{umr}(\eta)$ for each triple (u, m, r) in the training data \mathcal{D} . Incorporating $\Phi_{umr}(\eta)$ into the features of click-through rates weighted by β , which is drawn from a Gaussian distribution, gives the value of function $\Psi_{umr}(\Theta)$, using Equation (7.2). With the value of $\Psi_{umr}(\Theta)$, each preference relation o in \mathcal{D} can be obtained by the likelihood function defined as in Equation (7.4).

We learn the parameter $\bar{\Theta}$, consisting of η and β by fitting the probabilistic model TEM to the training data \mathcal{D} . Specifically, we obtain the posterior distri-

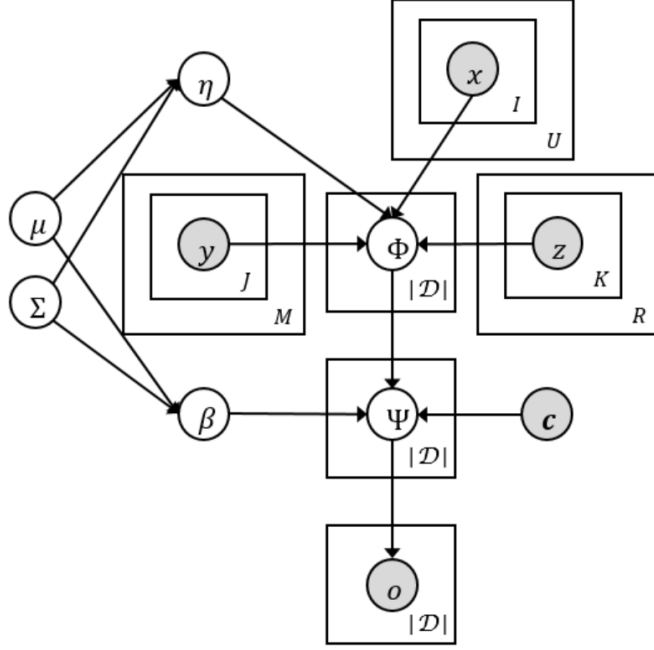


Figure 7.6: Graphical representation of Topic-specific Authority Analysis

tribution of the parameter Θ given all observations in training data \mathcal{D} , according to the Bayes' Rule:

$$p(\bar{\Theta}|\mathcal{D}) = \frac{p(\bar{\Theta})p(\mathcal{D}|\bar{\Theta})}{p(\mathcal{D})} \propto p(\bar{\Theta})p(\mathcal{D}|\bar{\Theta}), \quad (7.8)$$

where $p(\bar{\Theta})$ is the prior distribution defined as in Equation (7.7), and $p(\mathcal{D}|\bar{\Theta})$ is the likelihood of observing all preference relations defined as in Equation (7.5).

Maximum a posteriori (MAP) estimation is then conducted to infer the parameter $\bar{\Theta}$. That is, we find a $\bar{\Theta}$ such that the posterior probability $p(\bar{\Theta}|\mathcal{D})$ is maximized, i.e.,

$$\begin{aligned} & \arg \max_{\bar{\Theta}} p(\bar{\Theta}|\mathcal{D}) \\ &= \arg \max_{\bar{\Theta}} p(\bar{\Theta})p(\mathcal{D}|\bar{\Theta}) \\ &= \arg \max_{\bar{\Theta}} \left\{ \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\sum_i \bar{\theta}_i^2}{2\sigma^2}} \right. \\ & \quad \left. \times \prod_{(u,m,r_i,r_j) \in \mathcal{D}} \frac{1}{1 + e^{-g_{r_i,r_j}^{u,m} (\Psi_{umr_i}(\bar{\Theta}) - \Psi_{umr_j}(\bar{\Theta}))}} \right\}. \end{aligned} \quad (7.9)$$

We can equivalently transform this optimization problem into maximizing the logarithm of the posterior probability $p(\bar{\Theta}|\mathcal{D})$ as follows:

$$\begin{aligned}
& \arg \max_{\bar{\Theta}} \mathcal{L}(\bar{\Theta}) \\
&= \arg \max_{\bar{\Theta}} \log p(\bar{\Theta}|\mathcal{D}) \\
&= \arg \max_{\bar{\Theta}} \left\{ -\frac{\sum_i \bar{\theta}_i^2}{2\sigma^2} \right. \\
&\quad \left. + \sum_{(u,m,r_i,r_j) \in \mathcal{D}} \log \frac{1}{1 + e^{-g_{r_i,r_j}^{u,m}(\Psi_{umr_i}(\bar{\Theta}) - \Psi_{umr_j}(\bar{\Theta}))}} \right\}. \tag{7.10}
\end{aligned}$$

Equation (7.10) is an unconstrained convex optimization problem, which has a unique maximum. We use the *Limited-memory BFGS* algorithm [83] to solve the optimization problem and to estimate the parameters η and β . This involves computation of the gradients $\nabla_{\eta} \mathcal{L}(\bar{\Theta})$ and $\nabla_{\beta} \mathcal{L}(\bar{\Theta})$, i.e.:

$$\begin{aligned}
\frac{\partial \mathcal{L}(\bar{\Theta})}{\partial \eta_{ijk}} &= \sum_{(u,m,r_a,r_b) \in \mathcal{D}} \left\{ \frac{g_{r_a,r_b}^{u,m}}{1 + e^{g_{r_a,r_b}^{u,m}(\Psi_{umr_a}(\bar{\Theta}) - \Psi_{umr_b}(\bar{\Theta}))}} \right. \\
&\quad \left. \times x_{ui} y_{mj} (z_{r_{ak}} - z_{r_{bk}}) - \frac{\eta_{ijk}}{\sigma^2} \right\}, \tag{7.11}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathcal{L}(\bar{\Theta})}{\partial \beta_i} &= \sum_{(u,m,r_a,r_b) \in \mathcal{D}} \left\{ \frac{g_{r_a,r_b}^{u,m}}{1 + e^{g_{r_a,r_b}^{u,m}(\Psi_{umr_a}(\bar{\Theta}) - \Psi_{umr_b}(\bar{\Theta}))}} \right. \\
&\quad \left. \times (c_i^{umr_a} - c_i^{umr_b}) - \frac{\beta_i}{\sigma^2} \right\}. \tag{7.12}
\end{aligned}$$

With the parameter estimate $\hat{\Theta} = (\hat{\eta}, \hat{\beta})$, we can recommend a ranked list of entities r related to the main entity m searched by any user u . More specifically, given any triple (u, m, r) , we compute the value of function $\Psi_{umr}(\hat{\Theta})$ by:

$$\Psi_{umr}(\hat{\Theta}) = \sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K \hat{\eta}_{ijk} \cdot x_{ui} \cdot y_{mj} \cdot z_{rk} + \hat{\beta}^T \mathbf{c}^{umr}. \tag{7.13}$$

Related entities are then ranked in descending order of the $\Psi_{umr}(\hat{\Theta})$ scores. Entities with the highest scores will be recommended to the user u .

7.4.5 Three-way Interaction Effect

The TEM model raises the important question: How significant is the effect of the three-way correlations to modeling user clicks? To answer this question, one may test the statistical significance of the interaction effect with a t -test on the weights η which quantify the correlations among \mathbf{x}_u , \mathbf{y}_m , and \mathbf{z}_r . This practice, however, misinterprets the weight coefficients η in the nonlinear TEM model [5]. The correct measure of the three-way interaction effect for TEM should be a third partial derivative of the likelihood function $\mathcal{F}(\cdot)$ instead.

Let Δ denote either the derivative or the difference operator, depending on whether the corresponding feature values are discrete or continuous. The three-way interaction effect is then estimated by $\hat{\mu}_{xyz} = \frac{\Delta^3 \mathcal{F}}{\Delta x \Delta y \Delta z}$. When x , y and z are discrete features, the interaction effect can be derived as:

$$\begin{aligned}
 \hat{\mu}_{xyz} &= \frac{\Delta^3 \mathcal{F}}{\Delta x \Delta y \Delta z} \\
 &= \mathcal{F}(\eta_x + \eta_y + \eta_z + \eta_{xy} + \eta_{xz} + \eta_{yz} + \eta_{xyz} + \tilde{\mathbf{c}}) \\
 &\quad - \mathcal{F}(\eta_x + \eta_y + \eta_{xy} + \tilde{\mathbf{c}}) - \mathcal{F}(\eta_x + \eta_z + \eta_{xz} + \tilde{\mathbf{c}}) \\
 &\quad - \mathcal{F}(\eta_y + \eta_z + \eta_{yz} + \tilde{\mathbf{c}}) + \mathcal{F}(\eta_z + \tilde{\mathbf{c}}) \\
 &\quad + \mathcal{F}(\eta_y + \tilde{\mathbf{c}}) + \mathcal{F}(\eta_x + \tilde{\mathbf{c}}) - \mathcal{F}(\tilde{\mathbf{c}})
 \end{aligned} \tag{7.14}$$

where the η terms denote the weights of the features specified by the respective subscripts, and $\tilde{\mathbf{c}}$ represents the linear combination of all remaining features and weight coefficients. When some or all of x , y and z are continuous features, we can derive similar equations for the interaction effect, which are omitted due to the lack of space.

The standard error of the interaction effect estimate $\hat{\mu}_{xyz}$ is obtained by the Delta method:

$$\hat{\mu}_{xyz} \sim \text{Gaussian} \left(\mu_{xyz}, \frac{\partial}{\partial \eta} \left[\frac{\Delta^3 \mathcal{F}}{\Delta x \Delta y \Delta z} \right] \Omega_{\eta} \frac{\partial}{\partial \eta} \left[\frac{\Delta^3 \mathcal{F}}{\Delta x \Delta y \Delta z} \right] \right), \tag{7.15}$$

Table 7.2: Statistics of experimental datasets

Dataset	# users	# entities	# instances
Movie	36,641	15,409	224,567
Celebrity	26,371	2,016	1,450,609

which gives the estimate of the asymptotic variance of $\hat{\mu}_{xyz}$:

$$\hat{\sigma}_{xyz}^2 = \frac{\partial}{\partial \eta} \left[\frac{\Delta^3 \mathcal{F}}{\Delta x \Delta y \Delta z} \right] \hat{\Omega}_\eta \frac{\partial}{\partial \eta} \left[\frac{\Delta^3 \mathcal{F}}{\Delta x \Delta y \Delta z} \right], \quad (7.16)$$

where $\hat{\Omega}_\eta$ is a consistent covariance estimator of η .

For the t -test, we define the t statistic as $t = \frac{\hat{\mu}_{xyz}}{\hat{\sigma}_{xyz}}$. With the statistic, we test the null hypothesis that the overall effects of the three-way interactions equal zero for given training data, which gives p -value < 0.05 . So we reject the null hypothesis, which indicates the fact that the three-way interaction effect is statistically significant to modeling user clicks on related entities.

7.5 Empirical evaluation

In this section, we report the experimental results of TEM on real-world data collected by a commercial search engine. We compare the results of TEM against those of several competitors. Analysis and discussion of the experimental results are presented in this section.

7.5.1 Data

Although TEM is a generic probabilistic model which is applicable to recommending various kinds of entities, we take two specific types of recommendation tasks as case studies for empirical evaluation: *movie recommendation* and *celebrity recommendation*. The movie recommendation task is to recommend a ranked list of movies that are related to the movie searched by the user. For celebrity recom-

mentation, we aim to present to the user other celebrities related to the one he or she searched for.

We collected the entity pane log data for March 2013 through July 2013 from a commercial search engine. For the two recommendation tasks, movies and celebrities were extracted by aligning the entities in the log with those in Freebase. Freebase is a collaborative knowledge base of more than twenty million entities, including well-known people, places, movies and things. The basic statistics of the movie dataset and the celebrity dataset are given in Table 7.2.

Table 7.3 lists some features we used for the two recommendation tasks (Due to the space limitation, we do not list all the features here). Specifically, to develop user profiles, by joining *search click log*, *entity pane log* and the Freebase *knowledge base* (See Section 7.3 for details), we collected the popular entities the users had viewed together with their attributes/types as features, such as popular movies, pop stars and well-known writers. Each of the features was represented as the frequency of its occurrence in the logs for each user. In addition, for movie recommendation, with the help of the knowledge base we included the attributes (i.e., genres, countries, languages, etc.) of the movies viewed into the user dimension. This enables the model to learn user characteristics from the various aspects of their viewed movies, and thus to recommend related movies based on their preferences. As for celebrity recommendation, we included popular celebrities the users had viewed in the user-specific feature vectors, such as business leaders, musicians, actors and directors. For main entities and related entities, we constructed two different feature sets for the movie task and the celebrity task. For movie recommendation, we extracted the attributes of the movies as features, such as actors, directors, genres, languages, and subjects, whereas the celebrity recommendation model selected the celebrity-related features, such as the movies directed by the directors, the books written by the writers, and their spouses. With the domain-specific feature sets, TEM is able to discover the correlations

Table 7.3: Features for movie & celebrity recommendation

Movie recommendation		Celebrity recommendation	
<i>User dimension</i>	<i>Main & related movie</i>	<i>User dimension</i>	<i>Main & related celebrity</i>
Viewed entities		Viewed entities	Profession
Types of viewed entities	Actors	Types of viewed entities	Movie acted
Viewed movie's actors	Directors	Attributes of viewed entities	Movie directed
Viewed movie's directors	Genres	Viewed pop singers	Book written
Viewed movie's genres	Country of origin	Viewed business leaders	Music genre
Viewed movie's country	Language	Viewed writers	Organization
Viewed movie's language	Producers	Viewed musicians	Spouse
Viewed movie's producers	Series	Viewed actors	Nationality
Viewed movie's series	Story	Viewed film directors	Language
Viewed movie's story	Subject	Types
Viewed movie's subject	Music	
Viewed movie's music		
.....			

among the three dimensions for recommending related entities.

In addition to the features listed in Table 7.3, we obtained the features of click-through rates (CTR) which are considered very strong signals for recommendation. More specifically, based on the entity pane log, we collected r -specific CTRs: $CTR(r)$, (m, r) -specific CTRs: $CTR(m, r)$, and (u, m, r) -specific CTRs: $CTR(u, m, r)$. We also collected from the search log the frequency of entities viewed in the same session. As a result, for movie recommendation there were a total of 1653 features for each user, and 419 features for each main entity and related entity. For celebrity recommendation, there were a total of 1938 features for each user, and 562 features for each main entity and related entity.

7.5.2 Evaluation strategy

To evaluate the quality of entity recommendation, we split both the movie dataset and the celebrity dataset into a training set and a test set. The test set consisted of the latest page impression for each user, and the training set contained the rest. That is, TEM was used to rank a list of entities related to the last main entity

searched by each user.

Let Q be a set of tuples (u, m) . For each tuple $(u, m) \in Q$, a recommendation algorithm returns a ranked list of related entities with respect to user u and main entity m . To analyze the recommendation results, we used two evaluation metrics. The first metric was the standard Mean Reciprocal Rank (MRR). The Reciprocal Rank of a ranked list is the multiplicative inverse of the rank of the first hit in the list. The MRR score of a recommendation algorithm is the average reciprocal rank obtained by the ranked lists given by the algorithm with respect to the set Q . Formally,

$$MRR = \frac{1}{|Q|} \sum_{n=1}^{|Q|} \frac{1}{rank^{(n)}}, \quad (7.17)$$

where $rank^{(n)}$ is the rank of the first clicked entity in the ranked list for the n -th tuple. The other metric used for evaluation was called RankAcc. RankAcc was introduced to measure what fraction of preference orders $r_i \succ r_j$ is captured by a ranked list of recommended entities. Formally, we define RankAcc of a recommendation algorithm as:

$$RankAcc = \frac{1}{|Q|} \sum_{n=1}^{|Q|} \frac{|\{(r_i^{(n)}, r_j^{(n)}) | i < j \wedge r_i^{(n)} \succ r_j^{(n)}\}|}{|\{(r_i^{(n)}, r_j^{(n)}) | i < j\}|}, \quad (7.18)$$

where i and j are the ranks of related entities r_i and r_j in the ranked list, respectively. So $i < j$ suggests that entity r_i is ranked higher than entity r_j . Therefore, the fraction in Equation (7.18) gives the number of preference orders $r_i \succ r_j$ consistent with the rank orders out of the total number of pairs (r_i, r_j) induced by the rank.

7.5.3 Recommendation accuracy

We first evaluated the recommendation quality of the compared algorithms, *Random*, *Co-click*, *Production*, *CTR-model*, and *TEM* on the two real-world datasets. The *Random* approach is a naive algorithm which randomly ranks the related

entities in each page impression³. *Co-click* exploits the valuable signal that an entity should be recommended for another given entity if and only if the two entities are frequently co-clicked. Specifically, given a main entity m , *Co-click* estimates $p(r|m)$, the conditional probability of recommending related entity r , based on the number of their co-occurrences in the click log. *Co-click* then ranks the entities r by the conditional probabilities $p(r|m)$. The co-click signal by itself has been shown to be very effective and the strongest baseline method for entity recommendation [116]. *Production* represents the recommendation approach currently employed by a commercial search engine. It reflects the state of the art in the specific application by major search engines. *CTR-model* is a simplified version of *TEM*. It builds up the recommendation model in a way similar to *TEM*, except that *CTR-model* only utilizes the CTR features without incorporating the trilinear function $\Phi_{umr}(\eta)$. In essence, *CTR-model* and *TEM* build upon the same probabilistic framework, while different in feature sets used for training. We introduced the *CTR-model* in the interests of investigating the power of the CTR features derived from the entity pane log as well as the power of our probabilistic framework. We set $\sigma^2 = 5$ for the Gaussian prior in the probabilistic framework. For the *TEM* model, we set the number of random projection dimensions as 20, since that produced the best recommendations.

Figure 7.7 shows the MRR score of each algorithm for movie recommendation. From this figure, we observe that the other four methods are clearly superior to the *Random* approach. *Co-click* and *Production* give similar MRR results, as current search engines recommend related entities based on the co-click signal. It is interesting to see that *CTR-model* produces a high MRR result, even better than the state-of-the-art baseline *Production*. This shows the great potential of the click feedback in the entity pane log. Also, it suggests the ability of our

³The number of related entities presented in each page impression is greatly limited by a user’s screen size. It is normally ranging from 3 to 5. As a result, an algorithm which always provides the worst rankings would produce the MRR score approximately 0.25.

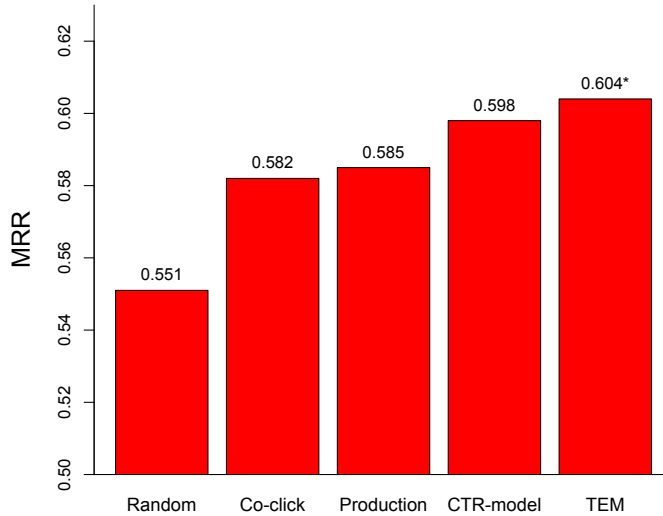


Figure 7.7: MRR for movie recommendation

probabilistic framework to leverage CTR signals for entity recommendation. To further compare *CTR-model* and *TEM*, we performed a paired t -test which had p -value < 0.05 , denoted by *. It indicated that the improvement of *TEM* over *CTR-model* is statistically significant. Figure 7.8 depicts the RankAcc scores of all compared algorithms for movie recommendation. From this figure, we observe the pattern similar to that of Figure 7.7.

For celebrity recommendation, the MRR and the RankAcc are shown in Figure 7.9 and Figure 7.10, respectively. Again, it is observed that *CTR-model* produces much higher MRR and RankAcc than those of both baselines *Co-click* and *Production*, and that *TEM* consistently outperforms all the other methods. To further measure the improvement of *TEM* over *CTR-model*, we performed a paired t -test between the two approaches. The ** in both figures indicate p -value < 0.01 , which show that *TEM* significantly improves over *CTR-model*.

7.5.4 Efficacy of personalization

Our *TEM* model personalizes recommendation results by taking the user dimension into consideration. The user dimension captures a user’s past interactions

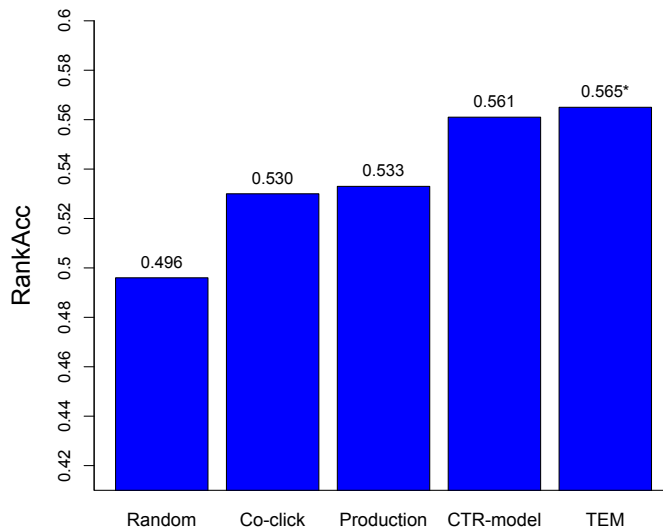


Figure 7.8: RankAcc for movie recommendation

with the search engine, such as the various types of entities he or she has viewed and their characteristics. *TEM* analyzes the underlying associations between induced user profiles and their actions on the entity pane to recommend the related entities tailored to their interests.

We took movie recommendation as a case study to investigate the personalization efficacy of *TEM*. Table 7.4 depicts an example of ranking the movie entities related to the movie *The Great Gatsby* by the four approaches *Co-click*, *Production*, *CTR-model*, and *TEM*. The particular search user was a fan of actor *Leonardo DiCaprio*, who starred in *The Great Gatsby*. Among the four related movies, the user jumped to *Django Unchained* to explore, which is the only movie starring *Leonardo DiCaprio*. Since the user had viewed the entity of *Leonardo DiCaprio*, by analyzing her historical logs *TEM* recognized her interest and thus put *Django Unchained* at the top of the ranked list. On the other hand, the other three approaches failed to customize the ranking of the related movies based on the user’s interest.

To further study the personalization efficacy of *TEM*, we conducted a quantitative evaluation. In particular, we first split the movie test set into five subsets

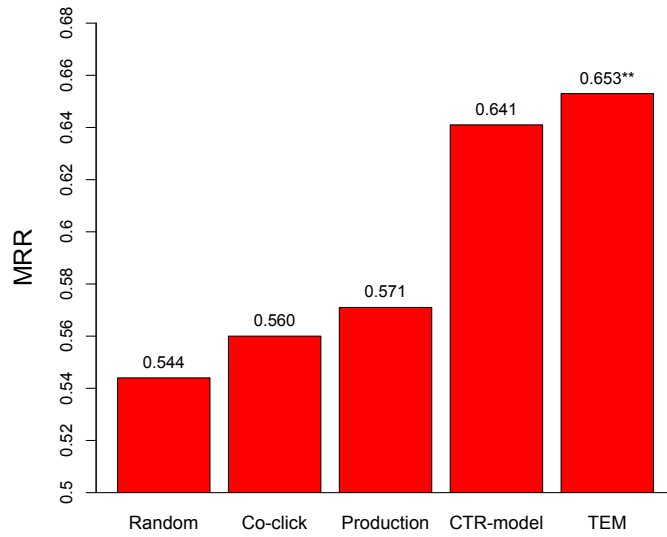


Figure 7.9: MRR for celebrity recommendation

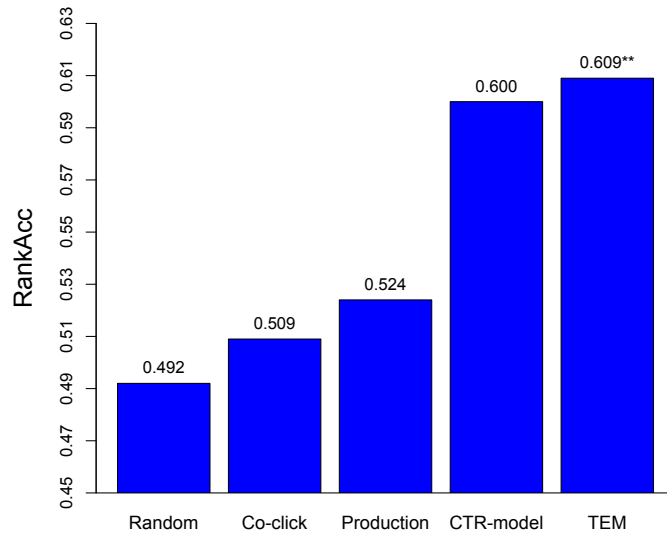


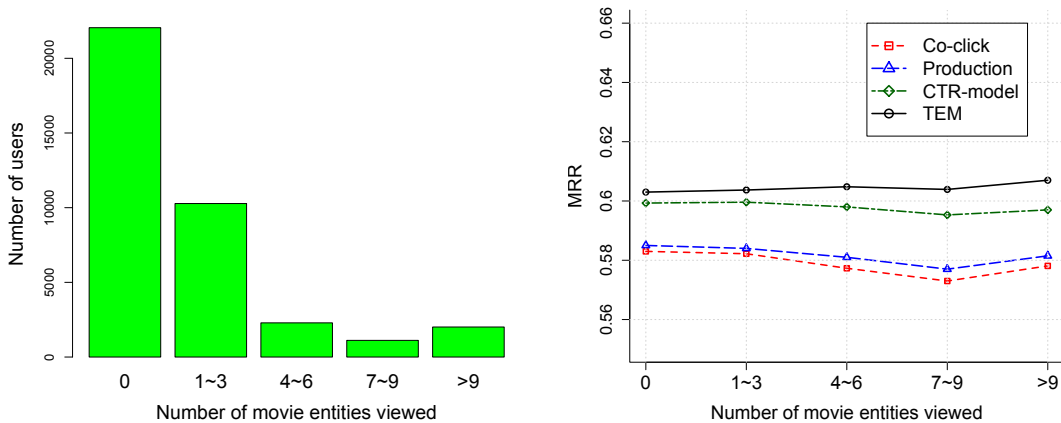
Figure 7.10: RankAcc for celebrity recommendation

Table 7.4: Related movies recommended for a fan of actor Leonardo DiCaprio by *Co-click*, *Production*, *CTR-model*, and *TEM*

User		
A fan of actor Leonardo DiCaprio		
Main movie entity		
The Great Gatsby		
Related movie entities		
<i>Co-click / Production</i>	<i>CTR-model</i>	<i>TEM</i>
Iron Man 3	Iron Man 3	Django Unchained
Man of Steel	Star Trek (2013)	Iron Man 3
Star Trek (2013)	Django Unchained	Star Trek (2013)
Django Unchained	Man of Steel	Man of Steel

based on the numbers of movie entities viewed by the users in the past. The number of users in each test subset is given in Figure 7.11(a). It is seen that 42% of users have viewed at least one movie entity in our log. The methods *Co-click*, *Production*, *CTR-model*, and *TEM* were used to recommend related movies for the users in each test set to evaluate their efficacy of personalization. Figure 7.11(b) shows the MRR scores of each algorithm on each test set. From this figure, it is observed that the three methods *Co-click*, *Production* and *CTR-model* produce consistent MRR results across the different test sets in spite of the drop for the “7~9” set⁴. This suggests that the unique preference of an individual user has little effect on the three methods for customizing the recommendation results. Our *TEM* model, however, increases the MRR as users have viewed an increasing number of movie entities. This confirms *TEM*’s ability to personalize recommended entities. Enriching user profiles will potentially improve the quality of recommendation of *TEM* for users.

⁴The MRR for the test set “7~9” is not statistically reliable given the small number of users in the set.



(a) User distribution over the numbers of movie entities viewed in the past (b) MRR for varying numbers of movie entities viewed in the past

Figure 7.11: User distribution and MRR

7.5.5 Effect of random projections

In this section, we investigate the effect of random projections on the quality of recommendation for *TEM*. In particular, we applied *TEM* to the training data of varying-dimensional feature space produced by random projections, and computed MRR for each random projection dimension. Figure 7.12 plots the MRR scores of the two recommendation tasks for different random projection dimensions. We observe that as more dimensions were used, *TEM* produced better recommendation results for both tasks. This is not surprising because increasing the number of random projection dimensions increases the capacity of the *TEM* model by giving it more tunable parameters, and also preserves more information about the original data.

7.6 Conclusion

This work addresses the problem of recommending entities related to the main entity returned to a user by a web search engine. We propose the probabilistic model *TEM*, which leverages the three data sources, *knowledge base*, *search click*

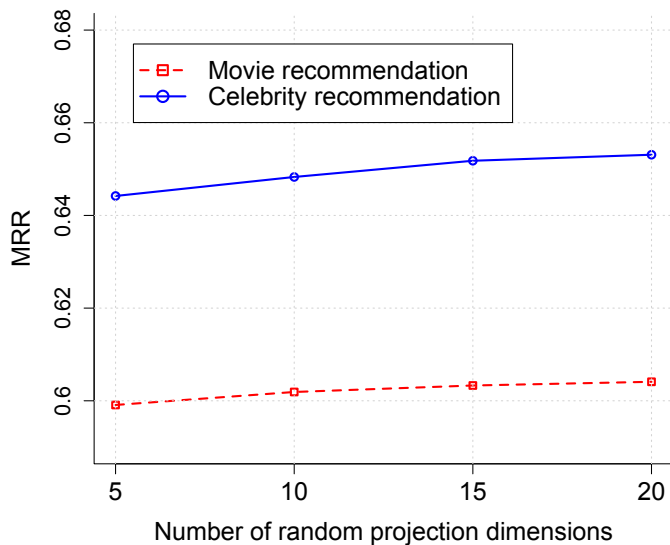


Figure 7.12: MRR for varying random projection dimensions

log, and *entity pane log*, for personalized recommendation of related entities. The TEM model not only utilizes the CTR signals derived from the entity pane log, but also exploits the three-way relationships among *user*, *main entity*, and *related entity*. Experimental results on movie recommendation and celebrity recommendation show that TEM with our probabilistic framework significantly improves over the state of the art technique employed by a major search engine. This confirms the effectiveness of TEM and the probabilistic framework on related entity recommendation.

CHAPTER 8

Conclusions and Future Work

This dissertation presents a family of Bayesian models specifically designed to analyze content and users in social media and web search engines. Here, we summarize the presentation of each model.

In Chapter 3, we introduce the FLDA model to characterize the topic-specific social influence of microblog users. FLDA incorporates the content of tweets and the network structure of microblogs into one unified model. Different from the previous work, such as Link-LDA, the FLDA model is specifically designed for microblogs in that it captures the fact that in reality a user sometimes follows another due to content-independent reasons. Moreover, in order to apply FLDA to a web-scale microblog network, we design a distributed Gibbs sampling algorithm for FLDA on the Spark distributed computing framework. Finally, the FLDA model is incorporated in a proposed general search framework for topic-specific key influencers, which provides a keyword search interface for users to freely query key influencers in different topic combinations.

In Chapter 4, we present two Bayesian nonparametric models, URM and UCM, to analyze the microblog data. Both models do not require the number of topics as an input parameter. Instead, they automatically determine the number of topics based on the observed microblog data. URM and UCM not only are able to integrate the analysis of tweet content and that of retweet behavior of users in the same statistical framework, but also jointly model users' interest in tweet and retweet.

In Chapter 5, we describe TAA which statistically models topic-specific authority. The TAA model properly captures the associations among users' interest and authority as well as the topics of favorited resources to exploit the signal of favorite clicks. The parameters in the TAA model are learned from a training set of observations constructed from two data sources: *sharing log* and *favorite log*. To overcome the limitation of missing negative feedback, we propose a preference learning technique embedding a new logistic likelihood function. An extension of typical collapsed Gibbs sampling is further proposed for Bayesian inference with the logistic likelihood.

In Chapter 6, we address the problem of inferring users traits – namely age, gender, religion and political view – from their search queries. We train our predictive models on a sample of Facebook users that have agreed to provide their Likes and other profile information for research purposes. We demonstrate that both Facebook Likes and search queries can be translated into a common representation via mapping to ODP categories. In addition, we address the data-shift problem by breaking up the problem into separate estimation tasks for demographics given category, and category given query history. For future work, we are interested in expanding the models to capture other types of user traits, such as personality, intelligence, happiness, or interests and measuring the applications of those inferred traits in personalization, reranking and monetization of the search results.

In Chapter 7, we propose another Bayesian model TEM, which leverages the three data sources, *knowledge base*, *search click log*, and *entity pane log*, for personalized recommendation of related entities. The TEM model not only utilizes the CTR signals derived from the entity pane log, but also exploits the three-way relationships among *user*, *main entity*, and *related entity*.

In addition to the tasks addressed above, Bayesian modeling can actually be used in a wide range of applications. For example, we've seen that both the FLDA model and the TAA model are able to recommend for users the key influencers or

experts relevant to their interest. The two models can be easily extended to build recommender systems in various domains in addition to social media.

Also, Bayesian modeling can be used to extract features for a learning model. For instance, we can append the posterior distributions of latent topics inferred from a Bayesian model as additional features. These new features provide informative signals about users unique topical interest, which can enhance the accuracy of the learning model.

Moreover, Bayesian modeling can be used to analyze usage data. Appropriate modeling of the usage data allows us to reveal underlying homogeneity and heterogeneity in usage behaviors of users. For example, it is able to identify multiple usage behaviors with the same latent intent, and meanwhile to properly captures the great diversity of the behaviors of users.

REFERENCES

- [1] Evrim Acar and Bulent Yener. Unsupervised multiway data analysis: A literature survey. *IEEE TKDE*, 2009.
- [2] Deepak Agarwal and Bee-Chung Chen. flda: Matrix factorization through latent dirichlet allocation. In *Proc. of WSDM '10*, pages 91–100, New York, NY, USA, 2010.
- [3] Amr Ahmed, Moahmed Aly, Joseph Gonzalez, Shravan Narayanamurthy, and Alexander J. Smola. Scalable inference in latent variable models. In *WSDM'12*, pages 123–132, 2012.
- [4] Jae-wook Ahn, Peter Brusilovsky, Jonathan Grady, Daqing He, and Sue Yeon Syn. Open user profiles for adaptive news systems: Help or harm? In *Proc. of WWW '07*, pages 11–20, Canada, 2007.
- [5] Chunrong Ai and Edward C. Norton. Interaction terms in logit and probit models. *Economics Letters*, 80(1):123 – 129, 2003.
- [6] Charles E. Antoniak. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, 11 1974.
- [7] Andrew Arnold, Ramesh Nallapati, and William W. Cohen. A comparative study of methods for transductive transfer learning. In *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops, ICDMW '07*, pages 77–82, 2007.
- [8] Yoram Bachrach, Michal Kosinski, Thore Graepel, Pushmeet Kohli, and David Stillwell. Personality and patterns of Facebook usage. In *Proceedings of the 3rd Annual ACM Web Science Conference, WebSci '12*, pages 24–32, Evanston, IL, 2012. ACM.
- [9] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. Topic-aware social influence propagation models. In *Proc. of ICDM '12*, pages 81–90, Washington, DC, USA, 2012.
- [10] Dominic Battré, Stephan Ewen, Fabian Hueske, Odej Kao, Volker Markl, and Daniel Warneke. Nephele/pacts: a programming model and execution framework for web-scale analytical processing. In *SoCC*, 2010.
- [11] Steven Bellman, Eric J. Johnson, Gerald L. Lohse, and Naomi Mandel. Designing marketplaces of the artificial with consumers in mind: Four approaches to understanding consumer behavior in electronic environments. *J. Interactive Marketing*, 20(1), 2006.

- [12] Paul N. Bennett, Filip Radlinski, Ryen W. White, and Emine Yilmaz. Inferring and using location metadata to personalize web search. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, SIGIR '11*, pages 135–144, Beijing, China, 2011. ACM.
- [13] Paul N. Bennett, Krysta Svore, and Susan T. Dumais. Classification-enhanced ranking. In *Proceedings of the 19th international conference on World wide web, WWW '10*, pages 111–120, Raleigh, NC, 2010. ACM.
- [14] Bin Bi and Junghoo Cho. Automatically generating descriptions for resources by tag modeling. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, CIKM '13*, pages 2387–2392, 2013.
- [15] Bin Bi, Ben Kao, Chang Wan, and Junghoo Cho. Who are experts specializing in landscape photography?: Analyzing topic-specific authority on content sharing services. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, pages 1506–1515, New York, NY, USA, 2014. ACM.
- [16] Bin Bi, Sau Dan Lee, Ben Kao, and Reynold Cheng. Cubelsi: An effective and efficient method for searching resources in social tagging systems. In *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering, ICDE '11*, pages 27–38, 2011.
- [17] Bin Bi, Hao Ma, Paul Hsu, Wei Chu, Kuansan Wang, and Junghoo Cho. Learning to recommend related entities to search users. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining, WSDM '15*. ACM, 2015.
- [18] Bin Bi, Lifeng Shang, and Ben Kao. Collaborative resource discovery in social tagging systems. In *Proceedings of the 18th ACM International Conference on Information & Knowledge Management, CIKM '09*, pages 1919–1922, 2009.
- [19] Bin Bi, Milad Shokouhi, Michal Kosinski, and Thore Graepel. Inferring the demographics of search users: Social data meets search queries. In *Proceedings of the 22nd International Conference on World Wide Web, WWW '13*, pages 131–140, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.
- [20] Bin Bi, Yuanyuan Tian, Yannis Sismanis, Andrey Balmin, and Junghoo Cho. Scalable topic-specific influence analysis on microblogs. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining, WSDM '14*, pages 513–522, 2014.

- [21] David Blackwell and James B. MacQueen. Ferguson distributions via polya urn schemes. *The Annals of Statistics*, 1(2):353–355, 03 1973.
- [22] Roi Blanco, Berkant Barla Cambazoglu, Peter Mika, and Nicolas Torzec. Entity recommendations in web search. In *International Semantic Web Conference (2)*, pages 33–48. Springer, 2013.
- [23] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003.
- [24] Vinayak R. Borkar, Michael J. Carey, Raman Grover, Nicola Onose, and Rares Vernica. Hyracks: A flexible and extensible foundation for data-intensive computing. In *ICDE*, 2011.
- [25] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *WWW'98*, pages 107–117, 1998.
- [26] Yingyi Bu, Bill Howe, Magdalena Balazinska, and Michael D. Ernst. Haloop: efficient iterative data processing on large clusters. *PVLDB*, 2010.
- [27] Robin Burke. Hybrid systems for personalized recommendations. In *Proc. of ITWP '03*, pages 133–152, Acapulco, Mexico, 2005.
- [28] David Carmel, Naama Zwerdling, Ido Guy, Shila Ofek-Koifman, Nadav Har'el, Inbal Ronen, Erel Uziel, Sivan Yogev, and Sergey Chernov. Personalized social search based on the user's social network. In *Proceedings of the 18th ACM conference on Information and knowledge management, CIKM '09*, pages 1227–1236, Hong Kong, China, 2009. ACM.
- [29] Deepayan Chakrabarti and Christos Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM COMPUTING SURVEYS*, 38(1):2, 2006.
- [30] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-graber, and David M. Blei. Reading tea leaves: How humans interpret topic models. In Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 288–296. Curran Associates, Inc., 2009.
- [31] Bee-Chung Chen, Jian Guo, Belle Tseng, and Jie Yang. User reputation in a comment rating environment. In *Proc. of KDD '11*, pages 159–167, New York, USA, 2011.
- [32] Ning Chen, Jun Zhu, Fei Xia, and Bo Zhang. Generalized relational topic models with data augmentation. In *Proc. of IJCAI '13*, pages 1273–1279, 2013.

- [33] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proc. of KDD '09*, pages 199–208, New York, NY, USA, 2009.
- [34] Wei Chu and Seung-Taek Park. Personalized recommendation on dynamic content using predictive bilinear models. In *Pro. of WWW '09*, pages 691–700, Madrid, Spain, 2009.
- [35] R. Coppi and S. Bolasco, editors. *Multiway data analysis*. North-Holland Publishing Co., Amsterdam, 1989.
- [36] Aron Culotta. Towards detecting influenza epidemics by analyzing Twitter messages. In *Proceedings of the First Workshop on Social Media Analytics, SOMA '10*, pages 115–122, Washington, DC, 2010. ACM.
- [37] Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Transferring naive Bayes classifiers for text classification. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 1, AAAI'07*, pages 540–545, Vancouver, BC, 2007. AAAI Press.
- [38] Hal Daumé, III and Daniel Marcu. Domain adaptation for statistical classifiers. *J. Artif. Int. Res.*, 26(1):101–126, May 2006.
- [39] Elena Erosheva, Stephen Fienberg, and John Lafferty. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences*, 101:5220–5227, 2004.
- [40] Michael Ettredge, John Gerdes, and Gilbert Karuga. Using web-based search data to predict macroeconomic statistics. *Commun. ACM*, 48(11):87–92, November 2005.
- [41] Wei Fan, Ian Davidson, Bianca Zadrozny, and Philip S. Yu. An improved categorization of classifier’s sensitivity on sample selection bias. In *Proceedings of the Fifth IEEE International Conference on Data Mining, ICDM '05*, pages 605–608, Washington, DC, USA, 2005.
- [42] Thomas S. Ferguson. A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, 1(2):209–230, 1973.
- [43] Sylvia Fruhwirth-Schnatter and Rudolf Fruhwirth. Data augmentation and mcmc for binary and multinomial logit models. In *Sta Mod Reg Str*, pages 111–132. 2010.
- [44] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. November 2013.

- [45] Saptarshi Ghosh, Naveen Sharma, Fabricio Benevenuto, Niloy Ganguly, and Krishna Gummadi. Cognos: crowdsourcing search for topic experts in microblogs. In *SIGIR '12*, pages 575–590, 2012.
- [46] Kevin Gimpel, Nathan Schneider, Brendan O’Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *ACL*, pages 42–47, 2011.
- [47] Kevin Gimpel, Nathan Schneider, Brendan O’Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT ’11, pages 42–47, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
- [48] Jeremy Ginsberg, Matthew H. Mohebbi, Rajan S. Patel, Lynnette Brammer, Mark S. Smolinski, and Larry Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–1014, February 2009.
- [49] Mark Girolami and Ata Kabán. On an equivalence between plsi and lda. In *SIGIR '03*, pages 433–434, 2003.
- [50] Sharad Goel, Jake M. Hofman, Sébastien Lahaie, David M. Pennock, and Duncan J. Watts. Predicting consumer behavior with Web search. *Proceedings of the National Academy of Sciences*, 107(41):17486–17490, October 2010.
- [51] Robert B. Gramacy and Nicholas G. Polson. Simulation-based regularized logistic regression. *Bayesian Analysis*, 7(3):567–590, September 2012.
- [52] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101:5228–5235, 2004.
- [53] Hiroshi Haramoto, Makoto Matsumoto, Takuji Nishimura, Francois Paneton, and Pierre L’Ecuyer. Efficient Jump Ahead for 2-Linear Random Number Generators. *INFORMS Journal on Computing*, 20(3):385–390, 2008.
- [54] Taher H. Haveliwala. Topic-sensitive pagerank: a context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):784–796, July 2003.
- [55] Gregor Heinrich. Parameter estimation for text analysis,. Technical report, University of Leipzig, 2008.

- [56] Thomas Hofmann. Probabilistic latent semantic analysis. In *In Proceedings of Uncertainty in Artificial Intelligence, UAI 99*, pages 289–296, 1999.
- [57] Chris C. Holmes and Leonhard Held. Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Analysis*, 1(1):145–168, March 2006.
- [58] Jian Hu, Hua-Jun Zeng, Hua Li, Cheng Niu, and Zheng Chen. Demographic prediction based on user’s browsing behavior. In *Proceedings of the 16th international conference on World Wide Web, WWW ’07*, pages 151–160, Banff, AB, 2007. ACM.
- [59] Yoshiyuki Inagaki, Narayanan Sadagopan, Georges Dupret, Anlei Dong, Ciya Liao, Yi Chang, and Zhaohui Zheng. Session based click features for recency ranking. In *Proc. of AAI ’10*. AAAI Press, 2010.
- [60] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. of STOC ’98*, pages 604–613, Dallas, Texas, USA, 1998.
- [61] Bernard J. Jansen and Lauren Solomon. Gender demographic targeting in sponsored search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’10*, pages 831–840, Atlanta, GA, 2010.
- [62] Akshay Java, Pranam Kolari, Tim Finin, and Tim Oates. Modeling the spread of influence on the blogosphere. In *WWW 2006 Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2006.
- [63] Rosie Jones, Ravi Kumar, Bo Pang, and Andrew Tomkins. ”I know what you did last summer”: query logs and user privacy. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, CIKM ’07*, pages 909–914, Lisbon, Portugal, 2007. ACM.
- [64] Pawel Jurczyk and Eugene Agichtein. Discovering authorities in question answer communities by using link analysis. In *Proc. of CIKM ’07*, pages 919–922, New York, 2007.
- [65] Andreas M. Kaplan and Michael Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business Horizons*, 53:59–68, 2010.
- [66] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *Proc. of RecSys ’10*, pages 79–86, Barcelona, Spain, 2010.

- [67] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 137–146, New York, 2003.
- [68] Eugene Kharitonov and Pavel Serdyukov. Gender-aware re-ranking. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 1081–1082, Portland, OR, 2012. ACM.
- [69] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *JACM*, 46(5):604–632, 1999.
- [70] Weize Kong, Yiqun Liu, Shaoping Ma, and Liyun Ru. Detecting epidemic tendency by mining search logs. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 1133–1134, Raleigh, NC, 2010. ACM.
- [71] M. Kosinski, P. Kohli, D. Stillwell, Y. Bachrach, and T. Graepel. Personality and website choice. In *Proceedings of the 3rd Annual ACM Web Science Conference*, WebSci '12, Evanston, IL, 2012.
- [72] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 420–429, 2007.
- [73] Lu Liu, Jie Tang, Jiawei Han, and Shiqiang Yang. Learning influence from heterogeneous social networks. *Data Mining and Knowledge Discovery*, 25:511–544, 2012.
- [74] Lori Lorigo, Bing Pan, Helene Hembrooke, Thorsten Joachims, Laura Granka, and Geri Gay. The influence of task and gender on search and evaluation behavior using google. *Inf. Process. Manage.*, 42(4):1123–1131, July 2006.
- [75] Grzegorz Malewicz, Matthew H. Austern, Aart J. C. Bik, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. Pregel: a system for large-scale graph processing. In *SIGMOD*, 2010.
- [76] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, NY, 2008.
- [77] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM*, 54(5), October 2007.

- [78] Alessandro Micarelli, Fabio Gasparetti, Filippo Sciarrone, and Susan Gauch. The adaptive web. chapter Personalized Search on the World Wide Web, pages 195–230. Berlin, Heidelberg, 2007.
- [79] Ramesh Nallapati and William W. Cohen. Link-plsa-lda: A new unsupervised model for topics and influence of blogs. In *Proceedings of the Second International Conference on Weblogs and Social Media*, 2008.
- [80] Ramesh M. Nallapati, Amr Ahmed, Eric P. Xing, and William W. Cohen. Joint latent topic models for text and citations. In *Proc. of KDD '08*, pages 542–550, New York, NY, USA, 2008.
- [81] Radford M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.
- [82] David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. Distributed algorithms for topic models. *J. Mach. Learn. Res.*, 10, December 2009.
- [83] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer series in operations research and financial engineering. Springer, New York, NY, 2nd edition, 2006.
- [84] Jahna Otterbacher. Inferring gender of movie reviewers: exploiting writing style, content and metadata. In *Proceedings of the 19th ACM international conference on Information and knowledge management, CIKM '10*, pages 369–378, Toronto, ON, 2010. ACM.
- [85] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *Proc. of WWW '98*, pages 161–172, Brisbane, 1998.
- [86] Aditya Pal and Scott Counts. Identifying topical authorities in microblogs. In *WSDM '11*, pages 45–54, 2011.
- [87] Marco Pennacchiotti and Ana-Maria Popescu. Democrats, Republicans and Starbucks aficionados: user classification in Twitter. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 430–438, San Diego, CA, 2011. ACM.
- [88] Nicholas G. Polson, James G. Scott, and Jesse Windle. Bayesian inference for logistic models using pólya-gamma latent variables. *JASA*, 108(504):1339–1349, 2013.
- [89] Ian Porteous. *Networks of Mixture Blocks for Non Parametric Bayesian Models with Applications*. PhD thesis, Long Beach, CA, USA, 2010. AAI3403449.

- [90] Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *KDD*, 2008.
- [91] Daniele Quercia, Michal Kosinski, David Stillwell, and Jon Crowcroft. Our Twitter profiles, our selves: Predicting personality with Twitter. In *PAS-SAT/SocialCom 2011*, pages 180–185, Boston, MA, 2011. IEEE.
- [92] Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proc. of WSDM '10*, pages 81–90, New York, USA, 2010.
- [93] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
- [94] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.
- [95] Lifeng Shang and Kwok-Ping Chan. A temporal latent topic model for facial expression recognition. In *Proceedings of the 10th Asian conference on Computer vision - Volume Part IV*, pages 51–63, 2011.
- [96] Donnavieve Smith, Satya Menon, and K. Sivakumar. Online peer and editorial recommendations, trust, and choice in virtual markets. *J. Interactive Marketing*, 19(3), 2005.
- [97] Alexander Smola and Shравan Narayanamurthy. An architecture for parallel topic models. *PVLDB*, 3(1-2):703–710, September 2010.
- [98] Micro Speretta and Susan Gauch. Personalized search based on user search histories. In *Proc. of WI '05*, pages 622–628, Washington, DC, USA, 2005.
- [99] Jian-Tao Sun, Hua-Jun Zeng, Huan Liu, Yuchang Lu, and Zheng Chen. Cubesvd: A novel approach to personalized web search. In *Proc. of WWW '05*, pages 382–390, Chiba, Japan, 2005.
- [100] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09*, pages 807–816, New York, NY, USA, 2009.
- [101] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):pp. 1566–1581, 2006.

- [102] Sergio Torres and Ingmar Weber. What and how children search on the web. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 393–402, Glasgow, UK, 2011. ACM.
- [103] L. R. Tucker. The extension of factor analysis to three-dimensional matrices. In *Contributions to mathematical psychology.*, pages 110–127. New York, 1964.
- [104] Twitter.com. Twitter turns six, 2012.
- [105] M. Alex O. Vasilescu and Demetri Terzopoulos. Multilinear image analysis for facial recognition. In *Proc. of ICPR '02*, pages 511–514, 2002.
- [106] Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In *KDD*, 2011.
- [107] Yu Wang, Gao Cong, Guojie Song, and Kunqing Xie. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *Proc. of KDD '10*, pages 1039–1048, New York, 2010.
- [108] Ingmar Weber and Carlos Castillo. The demographics of web search. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 523–530, Geneva, Switzerland, 2010. ACM.
- [109] Ingmar Weber, Venkata Rama Kiran Garimella, and Erik Borra. Mining web query logs to analyze political issues. In *Proceedings of the 3rd Annual ACM Web Science Conference*, WebSci '12, pages 330–334, Evanston, IL, 2012. ACM.
- [110] Ingmar Weber, Venkata Rama Kiran Garimella, and Erik Borra. Political search trends. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 1012–1012, Portland, OR, 2012. ACM.
- [111] Ingmar Weber and Alejandro Jaimes. Demographic information flows. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1521–1524, Toronto, ON, 2010. ACM.
- [112] Ingmar Weber and Alejandro Jaimes. Who uses web search for what: and how. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 15–24, Hong Kong, China, 2011. ACM.
- [113] Amy Tracy Wells and Lee Rainie. The internet as social ally. *First Monday*, 13(11), 2008.

- [114] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twiterrank: Finding topic-sensitive influential twitterers. In *Proc. of WSDM '10*, pages 261–270, New York, NY, USA, 2010.
- [115] Josh Jia-Ching Ying, Yao-Jen Chang, Chi-Min Huang, and Vincent S. Tseng. Demographic prediction based on users mobile behaviors. In *Mobile Data Challenge 2012 (by Nokia) Workshop*, Newcastle, UK., 2012.
- [116] Xiao Yu, Hao Ma, Bo-June (Paul) Hsu, and Jiawei Han. On building entity recommender systems using user click log and freebase knowledge. In *Proc. of WSDM '14*, pages 263–272, New York, NY, USA, 2014.
- [117] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 114–, Banff, AB, 2004. ACM.
- [118] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. In *NSDI'12*, 2012.
- [119] Jun Zhang, Mark S. Ackerman, and Lada Adamic. Expertise networks in online communities: Structure and algorithms. In *WWW '07*, pages 221–230, 2007.
- [120] Tong Zhao, Naiwen Bian, Chunping Li, and Mengya Li. Topic-level expert modeling in community question answering. In *SDM '13*, pages 776–784. SIAM, 2013.