1-2012

Predictive Modeling for Navigating Social Media

Meiqun HU
Singapore Management University, meiqun.hu.2008@smu.edu.sg

Citation

HU, Meiqun. Predictive Modeling for Navigating Social Media. (2012). 1-149.
Available at: https://ink.library.smu.edu.sg/etd_coll/79

PREDICTIVE MODELING FOR NAVIGATING SOCIAL MEDIA

HU MEIQUN

SINGAPORE MANAGEMENT UNIVERSITY

2011

Predictive Modeling for Navigating Social Media

by

HU Meiqun

Submitted to School of Information Systems
in partial fulfillment of the requirements for the Degree of
Doctor of Philosophy in Information Systems

Dissertation Committee:

LIM Ee-Peng (Advisor/Chair)
Professor of Information Systems
Singapore Management University

JIANG Jing (Co-advisor)
Assistant Professor of Information Systems
Singapore Management University

David LO
Assistant Professor of Information Systems
Singapore Management University

Christopher KHOO Soo Guan
Associate Professor, Division of Information Studies
Wee Kim Wee School of Communication and Information
Nanyang Technological University

Singapore Management University

2011

Abstract

Social media changes the way people use the Web. It has transformed ordinary Web users from information consumers to content contributors. One popular form of content contribution is social tagging, in which *users* assign *tags* to Web *resources*. By the collective efforts of the social tagging community, a new information space has been created for information navigation. Navigation allows serendipitous discovery of information by examining the information objects *linked* to one another in the social tagging space. In this dissertation, we study prediction tasks that facilitate navigation in social tagging systems.

For social tagging systems to meet complex navigation needs of users, two issues are fundamental, namely *link sparseness* and *object selection*. Link sparseness is observed for many resources that are untagged or inadequately tagged, hindering navigation to the resources. Object selection is concerned when there are a large number of information objects that are linked to the current object, requiring to select the more interesting or relevant ones for guiding navigation effectively. This dissertation focuses on three dimensions, namely the *semantic*, *social* and *temporal* dimensions, to address *link sparseness* and *object selection*.

To address link sparseness, we study the task of tag prediction. This task aims to enrich tags for the untagged or inadequately tagged resources, such that the predicted tags can serve as navigable links to these resources. For this task, we take a topic modeling approach to exploit the latent semantic relationships between resource content and tags.

To address object selection, we study the task of personalized tag recommendation and trend discovery using social annotations. Personalized tag recommendation leverages the collective wisdom from the social tagging community to recommend tags that are semantically relevant to the target resource, while being tailored to the tagging preferences of individual users. For this task, we propose a probabilistic framework which leverages the implicit social links between like-minded users, *i.e.* who show similar tagging preferences, to recommend suitable tags.

Social tags capture the interest of the users in the annotated resources at different times. These social annotations allow us to construct temporal profiles for the annotated resources. By analyzing these temporal profiles, we unveil the non-trivial temporal trends of the annotated resources, which provide novel metrics for selecting relevant and interesting resources for guiding navigation. For trend discovery using social annotations, we propose a trend discovery process which enables us to analyze trends for a multitude of semantics encapsulated in the temporal profiles of the annotated resources.

Contents

Appendices

List of Figures

List of Tables

List of Algorithms

Publications Arising from the Dissertation

Listed in reverse chronological order:

1. Meiqun Hu, Ee-Peng Lim and Jing Jiang. Social tag prediction for news pages: An empirical comparison between content-based and topic-based methods. Submitted to *Journal of the American Society for Information Science and Technology*.

2. Meiqun Hu, Ee-Peng Lim and Jing Jiang. Using social annotations for trend discovery in scientific publications. In Proceedings of the *Fifth Workshop on Human-Computer Interaction and Information Retrieval*, 2011.

3. Meiqun Hu, Ee-Peng Lim and Jing Jiang. A probabilistic framework to personalized tag recommendation. In *Proceedings of IEEE Second International Conference on Social Computing*, pages 33–40, 2010.

4. Meiqun Hu. A topic modeling approach to social tag prediction. In *Bulletin of IEEE Technical Committee on Digital Libraries*, 6(2), Fall 2010.

Acknowledgments

My time as a graduate student has spanned many years and across two schools. I am grateful to many people during these years who made this dissertation possible.

I would like to thank my advisor, Lim Ee-Peng, for his continuous support during these years. As an advisor, he has been actively involved in my research work and shared his knowledge and experiences. His advices kept me accurate about all the technical details. He has also given me endless encouragement, especially during challenging times. I am also grateful to have Jiang Jing as my co-advisor. Although her schedules are packed, she has always been punctual to our weekly discussions. She have provided valuable insights that keep me motivated and keep the work progressing.

I owe my deep gratitude to Christopher Khoo and David Lo for being my dissertation committee. Their stimulating suggestions on error analysis and run time measurements have strengthened this dissertation. Thanks must also go to Steve Miller for his affecting conversations at seminars and other gatherings. He has influenced me to envision this dissertation and research in general practical and useful for a wider audience.

I am indebted to Ms Ong Chew Hong for her administrative support during my studies at Singapore Management University. She serves with patience and cheering smiles. I couldn't imagine the days at our research center without her.

I also wish to thank faculty members at School of Computer Engineering, Nanyang Technological University, who provided me with generous support and encouragement when I started Ph.D research there in 2006. They include Huang Shell Ying (Associate Professor), Chang Kuiyu (Assistant Professor), Byron Choi (now with Hong Kong Baptist University), Sun Aixin (Assistant Professor), and Miao Chunyan (Associate Professor). Their teachings in the class, inquiries on my research work and comments given on my presentations have all helped improve my research and communication skills. I still benefit from their words today. I am also indebted to Ms Elain Koh (Secretary to Head of Division of Information Systems), Mr Chua Chiew Song (Supporting staff at Center for Advanced Information Systems) and Mr Lai Chee Keong (Supporting staff at Center for Advanced Information Systems), who offered administrative and technical support.

I am very glad that I had the chance to study in both NTU and SMU.

I would also like to express my deepest appreciation to my friends who have supported me during these years. Noi Sian, thank you for sharing your perspectives on research and recommending the doctoral consortium. Cane, thank you for believing and your kind words. Jing, Ling, Jun and Felipe, thank you for your care and warmth that cheers me up in many occasions. Clara, Stephanie Ong, Stephanie Lee and Kathleen, thank you for your unceasing encouragements.

I owe extra thanks to my family. To my mom and dad, for their love and everything. This dissertation is deicated to you. And most specially to my twin sister, Meishan, for pointing out the broken lines in my earlier writings, for her fantastic baking that made my days, and for being there with me every time.

My heartfelt thanks to all of you.

to my mom and dad

Chapter 1

Introduction

1.1 Motivation

The success of Web 2.0 in promoting user interactions and contributions has resulted in different types of social media for people to learn, play and communicate with others online. Among them is social tagging, also known as social bookmarking.

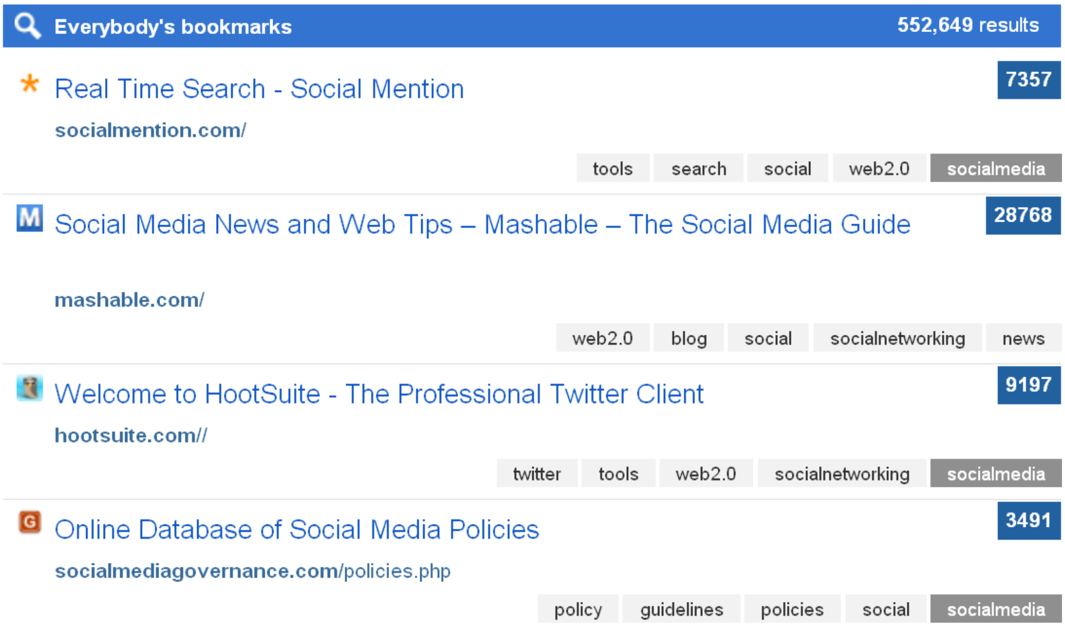
Social tagging refers to the practice of creating tags to annotate and organize Web resources in a shared, online setting. A tag is a freeform keyword not restricted to any predefined vocabulary, and it carries the essential meaning a user wants to assign to the annotated resource. By bookmarking and annotating Web resource with tags, users have a means to collect and categorize them. By storing these annotations at the social tagging sites, users are able to retrieve their bookmarks from any machine connected to the Web. Moreover, by exploring the bookmarks shared by other peers, users are able to find other interesting resources collectively annotated by the social tagging community.

Social tagging creates a new alternative information space for users to navigate vast amount of information. In this *information space*, the *information objects* are resources, users and tags, and these objects are linked via the assignment relationships, denoted by $\langle user, resource, tag : timestamp \rangle$. An assignment relationship is formed when a *user* uses a *tag* to annotate a *resource* at a certain *timestamp*. From the assignment relationships, *links* between pairs of information objects, *e.g.* user-tag, tag-resource and resource-user, can be extracted to support navigation. We regard *navigation* in the social tagging space as the process of traversing from one information object(s) to another, guided by the links from the current object(s) to the other object(s). At each navigation step, the current object(s) can be a resource, a user, a tag or a collection of multiple objects.

Navigation in social tagging systems is concerned with finding relevant and interesting information via the links between resources, users and tags. In most existing social tagging systems, navigation is supported by traversing the explicit links extracted from the assignment relationships between information objects, *e.g.* tag-resource etc. Given that the current object(s) at a navigation step may be linked to thousands, or even millions, of other objects, specialized navigational views can be provided to help users focus on a smaller set by selecting and ordering (or visualizing) the linked objects. For example, Delicious[1] provides **tag view** to show a combined view of all resources annotated with the given tag. In the tag view shown in Figure 1.1, resources are ordered based on the number of times the tag `socialmedia` has been assigned to them. Other example views provided by the existing social tagging systems are shown in Appendix A.

For social tagging systems to meet more complex navigation needs of users, two issues fundamental are to be addressed. The first issue is link sparseness, especially links for resources. Resources are navigable only if there are adequate links to them. The second issue is object selection. This may happen when a single object (*e.g.* tag) is linked to many other objects (*e.g.* resources or users). In the presence of object overloading, navigational views should select and present the significant objects, *e.g.* in terms of relevance and in-

[1] `www.delicious.com`



Everybody's bookmarks 552,649 results

Real Time Search - Social Mention 7357

socialmention.com/

tools search social web2.0 socialmedia

Social Media News and Web Tips – Mashable – The Social Media Guide 28768

mashable.com/

web2.0 blog social socialnetworking news

Welcome to HootSuite - The Professional Twitter Client 9197

hootsuite.com//

twitter tools web2.0 socialnetworking socialmedia

Online Database of Social Media Policies 3491

socialmediagovernance.com/policies.php

policy guidelines policies social socialmedia

Figure 1.1: An Example of Tag View

terestingness, with respect to the current object, helping users to navigate effectively. This dissertation focuses on studying prediction tasks that address the above two issues to facilitate navigation in social tagging systems. To elaborate these tasks, we identify three dimensions for link prediction and link selection, namely the *semantic*, *social* and *temporal* dimensions. Figure 1.2 depicts our conceptual model for navigating the social tagging space.

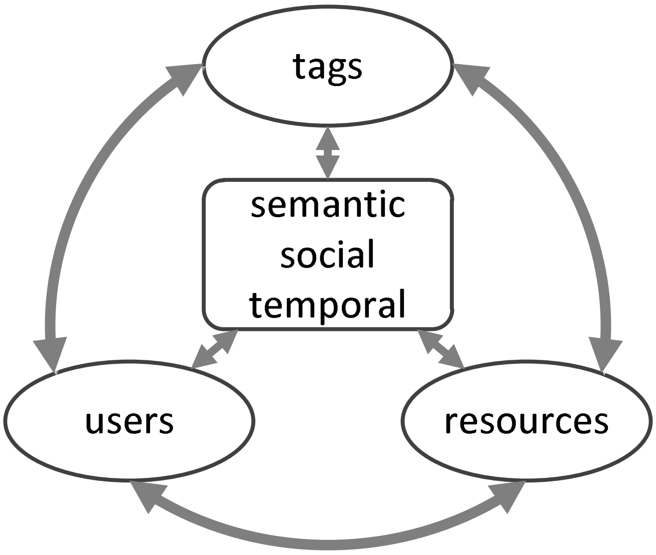


Figure 1.2: Three Dimensions for Navigational Links

In the *semantic* dimension, information objects are linked according to their underlying semantics [23]. For example, an article that discusses the release of iPad 2 is assigned tags such as `iPad2`, `tabletPC` and `technology`. In the existing social tagging systems, semantics are represented by individual tags, but the semantic dimension has not been fully exploited to link resources and tags. For example, suppose one understands that iPad2 is a tablet PC product, it is easy to infer the tag `tabletPC` for an article that was assigned the tag `iPad2`, despite that no user has assigned `tabletPC` to the particular article. Moreover, since social tags are freeform keywords created by different users, the same meaning may be represented in numerous forms including synonyms (*e.g.* `car` and `automobile`), acronyms (*e.g.* `world trade organization` and `WTO`) and morphological variations (*e.g.* `book` and `books`). Therefore, semantic groupings of related tags, such as topics, should be exploited to link resources and tags.

In the *social* dimension, information objects are selected based on the social relationships between users. For example, in tag recommendation, the tags recommended to users are those most frequently assigned by other users to the given resource. The act of bookmarking the same resource implies the *common interest* relationship between users. Such relationships *pervasively* exist between any two users as long as they bookmark the same resource. Another example is in resource recommendation, the resources recommended to users are selected from those bookmarked by the user's friends or group members. Friendships and group memberships specify the explicit *social links* between users, which may indicate common interest between the users. Recommending objects using explicit social links is based on the sociology theory known as *homophily*, which states that "users in the same group are like-minded, therefore the actions taken by a group member are likely to be adopted by other group members" [81]. In the existing social tagging systems, only pervasive and explicit social relationships are utilized for link selection, especially in the above recommendation tasks. However, there are other meaningful social relationships between users. For example, two users having similar tagging preferences may be like-minded, even though there is no explicit social links specified between them. Therefore, link selection should also exploit these implicit social relationships between users, *e.g.* for generating better recommendations to users.

In the *temporal* dimension, the information objects are selected or ordered according to their temporal attributes. For example, Delicious shows the most recent bookmarks to a particular resource on top of the earlier bookmarks. In the existing social tagging systems, timestamps of tag assignments are used to show point-wise attributes but not continuous trend. Furthermore, these point-wise attributes are not processed with respect to the different semantics of the information objects. Since tagging captures the interest of the users in the annotated resources at different times, one may construct temporal profiles for resources from their annotations over time. Such temporal profiles can then be analyzed to show interesting trends for the annotated resources. Trends with respect to the different semantic dimensions can be analyzed concurrently to help users further understand the relevance and recency of the annotated resources. For example, a resource on *Gibbs sampling*[2] may have been known to the statistical physics community for many decades. Yet, its relevance to the computer science community has only surfaced in recent years due to the introduction of Gibbs sampling techniques for parameter estimation in the latent Dirichlet allocation model[3] (LDA for short) [36]. When analyzed with respect to these two semantic dimensions, namely *statistical physics* and *computer science*, the temporal profile of this resource will likely demonstrate different recency of relevance. Therefore, trend analysis exploiting the temporal profiles of the information objects can provide useful information for selecting objects, hence guiding navigation.

[2] A sampling technique for parameter estimation in probabilistic graphical models.

[3] A probabilistic Bayesian model for text mining.

1.2 Research Overview

This dissertation focuses on three prediction tasks that analyze the semantic, social and temporal aspects of information objects to facilitate navigation in social tagging systems. Our research objectives are twofold. To address link sparseness, we study tag prediction task. To address object selection, we study personalized tag recommendation task and trend discovery task. In what follows, we further describe each task.

Tag prediction. Navigation to resources suffers from tag sparseness. Earlier studies on different social tagging systems [17, 41, 46, 47, 107] reported that, while a small amount of resources attracted extensive bookmarks, the vast majority are left untagged or inadequately tagged. The problem of sparseness is worsened by the fact that tags carrying the same meaning may appear in numerous forms [97]. The need to enrich tags for untagged or inadequately tagged resources is therefore critical. One way to enrich tags is to perform automated tag prediction. The predicted tags can then serve as navigable links to the resources.

Personalized tag recommendation. Users perform tagging primarily for personal consumption [78, 93, 120]. Specifically, users assign tags to organize resources within their personal bookmark collections, and relocate previously annotated resources through tags. Although the same meaning may be described using different tags by different users, individual users tend to be consistent in their own choice of tags even when other synonyms are present. This is because inconsistent tag assignments are ineffective for organizing and relocating resources within personal collections. Since resource organization is personal, tag recommendations should also be personalized. On one hand, tag recommendation should leverage the collective wisdom from the social tagging community to recommend tags that are semantically relevant to the resource. On the other hand, personalization should tailor to the tagging preferences of individual users.

Trend discovery. Social tags capture the interest of users in the annotated resources at different times. From these social annotations, one may construct temporal profiles for the annotated resource. By analyzing these temporal profiles, interesting trends may emerge. These emerging trends can be analyzed with respect to the different semantic dimensions encapsulated in the temporal profiles. Such trends unveil the non-trivial temporal attributes of the annotated resources, thus providing novel metrics for selecting relevant and interesting resources for guiding navigation. We identify trend discovery as a novel task that further facilitates navigation in social tagging systems. We demonstrate our vision by discovering the emerging topical trends in scientific publications.

1.3 Research Contributions

With respect to our research objectives, the contributions of this dissertation can be summarized as follows.

We study three prediction tasks that address link sparseness and object selection, which are major obstacles to effective navigation in social tagging systems. We propose holistic approaches to develop methods that address the challenges in these tasks. We empirically validate the effectiveness of our proposed methods using real-world datasets, and report the key findings from the empirical studies.

For tag prediction, we take a topic modeling approach to exploit the latent semantic relationships between resources and tags. We propose a probabilistic topic model, namely LDAtgg, which jointly models the content words and the social tags of Web documents. We develop a Gibbs sampling algorithm for learning the model parameters. We adopt Bayesian inference for estimating the probabilities of candidate tags for the untagged Web documents. We conduct experiments on a novel collection of real-world tagging data crawled from

Delicious. Our evaluation using this dataset shows significant advantage of LDAtgg model over the strongest baseline methods in prediction accuracy. We further analyze the prediction results with respect to a number of characteristics of tags and documents to give deep insights of the various tag prediction methods and the dataset. These characteristics include adequacy of the ground truth tags (*i.e.* the amount of tags assigned to the test document), frequency of tags (*i.e.* the number of times tags are seen in the corpus) and obviousness of tags (*i.e.* whether tags are also content words in the corresponding test document).

For personalized tag recommendation, we propose a probabilistic framework which leverages on the implicit social links between users, *i.e.* like-minded users who show similar tagging preferences, so as to find relevant tags for recommendation. We address personalization by modeling the probabilistic tagging preferences of individual users, namely *personomy translations*. We devise distributional similarity measures, such as Jensen-Shannon divergence and L1-norm, for comparing such tagging preferences between users, so as to perform *neighbor-based translations*. We conduct experiments on a benchmark dataset from BibSonomy, which has been used in the ECML PKDD[4] Discovery Challenge 2009. We evaluate the recommendation accuracy of the proposed *neighbor-based translation* framework across a range of similarity measures for finding neighbors (*i.e.* like-minded users), with variations on the formulation of tagging preference patterns. Our experimental results show that neighbor-based translations (trans-n) have clear advantage over their target-user-solely counterparts (trans-u) under the same framework. We further analyze the parameters tuned for individual users, which gives us quantitative understanding on *personal preference vs. social influence* for individual users.

For trend discovery, we propose a trend discovery process which enables us to analyze trends for a multitude of semantics encapsulated in the temporal

[4] The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases

profiles of the annotated resources. We perform topic modeling to capture the semantics in the temporal profiles. We propose trend estimation methods, such as the *sigmoid estimator*, to effectively parameterize these temporal profiles, supporting trend comparison and trend selection in terms of *emergence amplitude*, *ruling gradient* and *emergence time*. For empirical study, we include social tags and citations as two forms of social annotations for scientific publications. We conduct experiments on real-world datasets from CiteULike (for social tag annotations) and ACM Digital Library (for citation annotations). We perform a range of trend analysis tasks using the proposed trend discovery process, aiding researchers and information seekers to understand the impact of individual publications as well as the annotation community on the whole. We demonstrate the ability to select emerging publications for a given topic and to select emerging topics for a given publication using the proposed trend discovery process.

1.4 Dissertation Organization

The subsequent chapters of this dissertation are organized as follows:

- Chapter 2 summarizes the existing studies on social tagging systems, including the role of social tags and folksonomies for information organization and the use of social tags for resource navigation. Following these, the chapter also provides an overview of the existing studies on prediction tasks closely related to ours.

- Chapter 3 formulates the tag prediction task, and describes the probabilistic topic model we proposed for solving the task. We conduct experiments on news articles annotated on Delicious, and discuss our findings using the dataset.

- Chapter 4 focuses on the personalized tag recommendation task. We present our proposed probabilistic framework for solving the task, and our empirical study on the benchmark dataset for the ECML PKDD 2009 Discovery Challenge.

- Chapter 5 introduces the trend discovery task for providing novel metrics for link selection. We demonstrate the emerging trends found for scientific publications using social tags and citations as two forms of social annotations.
- Chapter 6 concludes the thesis by summarizing our contributions, main findings, limitations on the current work, and gives our suggestions for future research.

Chapter 2

Literature Review

Social tagging research can be broadly classified into the following three areas [111]:

(i) Tag-based information organization, which compares social tags with traditional metadata for organizing and accessing information;

(ii) Tagging behaviors and dynamics, which studies the motivations for tagging, the types of tags being generated, and the statistics of tagging activities;

(iii) Enhancement to social tagging systems, which consumes tagging data to facilitate better navigation, such as visualizing and ranking resources, tags, and users.

The prediction tasks we study in this dissertation address the third area.

In this chapter, we first provide an overview of studies in the first two areas, and highlight the well-known findings, which lay the foundation for our research. We then review previous studies that address useful and interesting Web mining tasks by consuming tagging data. Following that, we present a comprehensive summary of the existing approaches to tag prediction, personalized tag recommendation, and trend discovery, which are the three main research tasks of this dissertation. We also present a summary of topic model research, since topic modeling is adopted intensively in our tag prediction and trend discovery tasks.

2.1 Tag-based Information Organization

Tags have been regarded as an alternative to metadata (*i.e.* data about data) in organizing and accessing information. Traditionally, metadata is created by professional catalogers and librarians, who are specially trained for the task. While professionally created metadata are of high quality, the small community of such information professionals could not cope with the explosive growth of Web content. An alternative solution to this is to let the authors create metadata for their own content, such as bloggers labeling Weblog posts. This approach does not work well as not all authors are willing and capable of assigning good quality metadata. Furthermore, the approach has left out many other users who are interested in the content. Social tagging is therefore introduced to engage all users in the process of creating metadata for Web content. However, when compared with traditional metadata (with controlled vocabulary, complex rule sets and ontologies), tag-based information organization has both advantages and disadvantages.

The disadvantages of social tags for information organization include: (a) *polysemy*: the same tag may be used to refer to different concepts [73, 80]; (b) *synonymy*: the same concept may be described using different tags by different users [73, 80]; (c) *lexical variants*: the same word root may appear in different forms for the same concept, such as singular *vs.* plural forms, verb conjugations, acronyms, etc. [73, 80]; (d) *misspellings and errors*; (e) *lack of precision*: tags are too general, hence lack discriminative power [24, 73]; and (f) *lack of hierarchy*: the folksonomy, that results from social tagging, is a flat classification system, where tags are descriptors [73].

The advantages of social tags for information organization include: (a) it is based on *the conceptual model of users*: tags define the relationships between the online resources and concepts in the users' mind [39, 80]; (b) *low barriers and low cognitive costs*: no training or prior knowledge is required to contribute tags [80]; (c) *findability*: users invent personally meaningful tags, easing tasks such as re-finding resources [99]; (d) *self-normalization*: the collective vocabulary becomes more consistent over time, without external controls [34, 101]; and (e) *social navigation*: folksonomy supports serendipitous discovery through browsing other users' bookmark collections [80].

Given the advantages and disadvantages of folksonomy for information organization, a number of studies have tried to assess how effective tags can be used for searching.

Lin *et al.* [70] reported three empirical studies on the characteristics of social classification, comparing social tags with controlled vocabularies and title-based automatic indexing. They observed little overlap among terms derived from these three indexing methods, but tags are more similar to automatic indexing than to controlled vocabulary indexing. Their study also suggested that tags can be categorized into meaningful and stable groups to improve tag-based searching and browsing.

Al-Khalifa and Davis [1] examined the relationship between folksonomy tags and keywords extracted by automated indexing systems. They employed expert indexers to evaluate the quality of folksonomy tags *vs.* machine-generated keywords, and the percentage of overlap between these two kinds of index terms and human-generated keywords were calculated respectively. Folksonomy tags overlapped significantly with human generated-keywords, in contrast to the automatically generated ones. Furthermore, the professional indexers preferred the semantics of the tags over the automatically extracted keywords.

Smith [104] studied LibraryThing tags and the subject headings assigned to a small sample of books and found that the tags identify latent subjects accurately. Bischoff *et al.* [7] studied tagging data in different kinds of resources and systems, including Web pages (Delicious), music (Last.fm), and images (Flickr). They concluded that for the three social tagging systems, tags add new information to the resources, and a large proportion of tags are accurate and reliable. Most tags can be used as search terms and in most cases tagging behavior exhibits approximately the same characteristics as searching behavior. Based on these observations, they concluded that tags, found in different social tagging systems, could improve searching.

Chi and Mytkowicz [20] proposed an information theoretic approach to tag quality assessment, defined as the reduction in entropy in retrieving a particular document using a tag. They observed that the mutual information between tags and documents is linearly decreasing with time, so tags are becoming less and less useful as search terms.

Most of the studies agree that: (i) folksonomy tags exhibit a high degree of similarity to subject descriptors and controlled vocabularies used by experienced human indexers, although the indexing behaviors of users and professionals are different; and (ii) social tagging is unlikely to replace conventional knowledge organization systems, but offers an alternative way to develop vocabularies for supporting information access.

2.2 Tagging Behaviors and Dynamics

In this section, we focus on two groups of studies, namely the users' motivations in tagging and the evolving dynamics of the social tagging communities.

Users' Motivations in Tagging

Tagging is considered a means of sense-making by individual users. The primary benefits from tagging are self-organization and re-discovery [93]. Since tags are publicly visible to other users, they create social awareness that motivate users to tag.

Ames and Naaman [4] presented a taxonomy of tagging motivations in sociality and function dimensions: self *vs.* social, and organization *vs.* communication. They surveyed users of ZoneTag, a mobile application dedicated to tagging Flickr photos, and found that the participants were mainly motivated to organize resources by tagging for themselves as well as the general public. Similar findings were echoed in the work by Nov *et al.* [89], in which quantitative evaluations were conducted. They found that tagging activity levels were positively correlated with the self and public motivations, but were not as strongly correlated with family and friends motivations.

Marlow *et al.* [78] also summarized the motivations for users to tag movies, including: for contribution and sharing (*e.g.* communication), for expressing opinions (*e.g.* recommendation to others), for getting attention, for self presentation, for performance and activism [78], and for enabling other functionality (*e.g.* profile creation [99]).

This multitude of motivations has resulted in various types of tags for annotating resources. Based on data in Delicious, Golder and Huberman [34] identified seven types of tags:

For identifying what (or who) it is about Including common nouns and proper nouns, *e.g.* people or organizations;

For identifying what it is Such as `article`, `blog` and `book` etc;

For identifying who owns it Such as the owner of the Weblog post;

For refining existing categories Including numbers, which are often co-assigned with other tags, *e.g.* `2008` co-assigned with `election`;

For identifying qualities or characteristics Including most adjectives *e.g.* `scary`, `funny`, `stupid`, `inspirational` etc;

For self-referencing by the annotator *e.g.* `mystuff` and `mycomments`;

For organizing tasks Such as `toread` or `jobsearch`;

Dynamics of Social Tagging Community

A number of studies found that social tagging data often follow power law distribution [34, 96, 119]. Wetzker *et al.* [119] found that power law distributions were shown in relationships such as users-resources, tags-resources, and bookmarks-resources. Based on a large dataset from Delicious, they observed: (i) the top 1% of users generated 22% of all bookmarks, and the top 10% generated 62%; (ii) 700 tags (among 7 million) accounted for 50% of all assignments; (iii) 39% of all bookmarks are dedicated to the top 1% of URLs and 61% to the top 10%. They further noted that Delicious was dominated by technologically sophisticated users [119].

Studies also found a remarkable stability in the tagging activities [34, 99]. Golder and Huberman [34] presented numerous quantitative analyses, also focused primarily on data from Delicious. They noted that 67% of the URLs reached their peak popularity within 10 days of appearing on Delicious. They also found that the proportion of frequencies of tags for a given URL stabilized after around being annotated 100 times.

Robu *et al.* [96] found that there was an implicit form of "consensus" from different users annotating the same resource. They observed that the final tag frequencies for most resources converged to power law distributions, and the most used tags best describe the annotated resource.

While the above studies have taken the *macro view* on the social tagging community as a whole, there are also studies taking the *micro view* to characterize individual users and sub-communities. Marvasti and Skillicorn [79] found little evidence of user communities among users using a given tag, and little evidence of similarities among the documents tagged by a given user. Their analysis suggested that each individual tends to have wide interests, as expressed in what they tag. Körner *et al.* [60] proposed several measures for examining the degree of contribution by individual users. They distinguished *categorizers* from *describers*. The former typically uses a small set of tags as a replacement for hierarchical classification schemes, whereas the latter annotates resources with a wealth of freely associated, descriptive keywords. They found that *verbose* users (*i.e.* describers) were most useful for the emergence of tag semantics. They further noted a subset containing only 40% of the most verbose taggers can produce results that match and even outperform the semantic precision obtained from the whole dataset.

Sen *et al.* [99] examined factors that influence individual users' choice on tags and the degree to which community members share a vocabulary. Within the context of MovieLens[1], a movie recommendation system, they presented a quantitative model to study factors such as personal tendency (*e.g.* personal interest and knowledge) and community influence (*e.g.* tags used by other users). They found that prior exposure to a tag did influence individual's tag selection, leading towards consistency in tag vocabulary.

2.3 Applications Consuming Tagging Data

Hayes and Avesani [42] examined how tags can be used to cluster bloggers and their blog posts. Their empirical study on 13,518 blogs suggested that tags can be regarded as gold standard for cluster coherence. Based on the intuition that *tags reflect the interests of users*, Li *et al.* [67] proposed to group users and resources by topics of interests mined from tagging data. Kashoob *et al.* [58] used tags to discover latent communities of users. Yin *et al.* [121] utilized tag features to represent Web objects, and found improved performance in their Web object classification task.

Among many applications, tagging data is mostly explored in recommender systems, where tags are used to derive user profiles as well as resource descriptors, and these profiles and descriptors may be matched for better resource recommendation.

Tso-Sutter *et al.* [112] regarded tags as *local descriptions* to the items by

[1] `www.movielens.org`

the users. They proposed to combine item-based and user-based collaborative filtering approaches in the item recommendation task. Tags served as additional links between items and users other than rating data. They decomposed the three-dimensional relations between items, users and tags into three two-dimensional relations, namely item-user, item-tag and user-tag. Their experiments on article recommendation in CiteULike showed that, when tags are incorporated, the proposed item recommendation method yield significantly higher accuracy than the baseline methods.

To perform article recommendation on CiteULike, Parra and Brusilosky [92] explored the use of tags for finding neighbors for user-based collaborative filtering. They proposed a modified version of the BM25 model based on users' tags for computing the similarity between users. They suggested that tag-based BM25 can be considered an alternative to Pearson correlation based on users' ratings. Sen *et al.* [100] described a preference inference algorithm based on users' annotations on movies, which leverages tags with users' ratings to recommend movies. They examined 11 signals of user's preferences on tags, including tag searches, with respect to item ratings. They found that a hybrid tag-preference and collaborative filtering algorithm gave strong performance for both the prediction (*i.e.* to predict ratings to items) and recommendation (*i.e.* to recommend a list of items) tasks.

Traditional recommender systems suffer from the *cold start* problem, in which there is too little information about the new user or the new resource to make recommendations. Bogers and van den Bosch [13] suggested that it took about two years for the cold start problem to disappear in the case of CiteULike. Studies leveraging social tags for item recommendation and their findings on improvement recommendation accuracy are encouraging in the sense that, user-created tags provide useful information for profiling users as well as items to perform collaborative filtering for item recommendations.

2.4 Summary of Tag Prediction Research

Tag prediction aims at enriching tags for Web resources that are untagged or inadequately tagged. In our discussion, we refer to the resource for which tag are to be predicted as the *target resource*. We categorize the existing approaches to tag prediction into the following seven categories:

Content-based approach extracts and selects keywords or keyphrases from the text content of resources as predicted tags [22, 82, 87]. To automatically select keyphrases from the text content of resources, Medelyan *et al.* [82] proposed a two-stage method, which consists of *keyphrase selection* and *keyphrase filtering*. To select candidate keyphrases, they extracted n-grams (a sequence of consecutive n terms) from the resource content based on the well-known Kea system [27]. They adopted $n \leq 3$. To filter candidate keyphrases, they performed binary classification on them using semantic features derived from a Wikipedia corpus. These features include *term frequency*, *inverse document frequency*, *position of the first occurrence*, *the distance between its first and last occurrences*, *how often a candidate keyphrase appears as tags*, *length of the candidate keyphrase*, *the degree to which the candidate keyphrase is semantically related to other candidate keyphrases*, *the probability of the candidate keyphrase appearing as anchor text in Wikipedia* (also called *Wikipedia linkage*), and *inverse Wikipedia linkage*. Murfi and Obermayer [87] proposed a concept-based keyword extraction method, where *concepts* are latent dimensions relating keywords and documents. They learned keyword-concept relationships and concept-resource relationships, and the candidate tags are selected from the concept(s) rather than directly from the resource content. They found that allowing multiple concepts for each document gave better performance than their single-concept based counterparts. The document-concept-keyword relationship in their methods can be compared to the document-semantic-term relationship in latent semantic indexing [48] and the document-topic-word relationship in latent Dirichlet allocation [12, 30] for modeling the resource content. Diaz-Aviles *et al.* [22] proposed to predict tags for each target resource by selecting content words from an ad hoc Web corpus, where each ad hoc Web corpus is specifically constructed for the target resource by query the Web using content words of the target resource and collecting the content of the top results returned. Content-based approach requires the target resources to contain text content, hence, is not easily applicable to multimedia resources, such as images and videos. Moreover, this approach assumes that tags originate from the content of the resources, it does not model the tag vocabulary separately from the word vocabulary. Content-based approach, on the other hand, can predict novel tags not seen in the existing tag vocabulary. In other words, this approach is capable of identifying relevant novel tags from the resource content.

Topic-based approach develops probabilistic topic models, *e.g.* by extending latent Dirichlet allocation (LDA for short), to model the generation process for tags. This approach assumes that, each resource covers a number of latent topics, and each tag is generated from one of the latent topics covered by the respective resource. Such generation process is described as sampling a term from the multinomial distribution over the tag vocabulary. The latent topics differ in their multinomial distributions, *i.e.* one topic may have a higher probability of generating some tags over other tags, while other topics do otherwise. For tag prediction, the model first estimates the likelihoods of the latent topics covered by the target resource, and then computes the posterior probabilities of the candidate tags given the likelihoods of the topics. Often, the likelihoods of the latent topics for each resource is learned from the content words of the resources. Krestel *et al.* [61, 62] modeled topics on tags exclusively, *i.e.* without considering content words. Since no resource content is modeled, estimating the topic mixture for the target resource must rely on the existing tags of the target resource. In other words, their approach is limited to enriching tags for the (inadequately) tagged resources. This same limitation was found in the rule-based approach described earlier. A number of studies, including ours, exploit topic-based approach considering both tags and content words of the resources [18, 22, 71, 102]. These existing methods differ in: (i) modeling the correlation between the topics for tags and the topics for words, such as the correspondence assumption [71] and the conditionally independent assumption [102]; and, (ii) modeling the additional information available for the type of target resources, such as the authors of the underlying resources [18] and the users' preference of assigning tags [71].

Language-based approach develops probabilistic models, following the language modeling framework in information retrieval, to capture the relationship between tags and content words of the resources. While language models for information retrieval compute the likelihood of the query being generated by each document, language-based approach for tag prediction computes the likelihood of the tag being generated by the target resource. Different from information retrieval, tags and content words of the resource do not share the same vocabulary, unlike query keywords. In the training phase, the model estimates a joint probability distribution of the set of tags and the set of words from the same resource. In the prediction phase, the model estimates the posterior probabilities of the candidate tags given the observed words of the target resource. Different from topic-based approach, no latent topic is introduced to relate tags with content words. Givon and Lavrenko [31] studied *shelf labels* prediction for books in Goodreads[2], where the shelf labels are assigned collectively by the user community. They adopted relevance model, a

[2] www.goodreads.com

relevance weighting technique under the language modeling framework, for capturing the dependencies of the *bags of labels* on the *bags of words* of the book descriptors. Sun *et al.* [108] studied tag prediction for scientific publications in CiteULike[3]. In addition to language modeling, they also incorporated link-based approach, where links are induced by content similarity, and content-based approach for post-hoc selection on the candidate tags.

Link-based approach predicts tags from tags assigned to other resources that are linked to the target resource [5, 17, 72, 97, 107]. This approach is based on the hypothesis that *linked resources are likely to share common tags*. Some previous methods based on this approach rely on explicit links between resources, *e.g.* hyperlinks between Weblogs [107] and citation links between scientific publication [17]. Since such explicit links may not be available for all types of resources, some work also exploited the use of implicit links between resources for propagating tags, *e.g.* based on content similarity between resources [5, 72, 97]. Au Yeung *et al.* [5] found that tag prediction based on user-induced links is significantly more accurate than those based on existing hyperlinks. However, irrelevant tags may also be erroneously propagated. Sarmanto *et al.* [97] discussed two types of errors when propagating tags (named as topic labels in their setting) for news feeds, namely *mis-propagated location labels* and *incorrect but relevant entity names*.

Rule-based approach infers new tags from the existing tags of the target resource using tag co-occurrence rules [6, 47, 84]. Such co-occurrence rules are mined using association rule mining techniques, using the set of tags assigned to the same resource as a transaction record. When applied to tag prediction, existing tags (*i.e.* tags already assigned to the resource) of the target resource are used to match the antecedent of

[3] `www.citeulike.org`

the rules, and the consequents of the applicable rules are the candidate tags. Often, the confidence of the applicable rules are used to weight the candidate tags. Heymann *et al.* [47] found that, this approach can give very high precision. Yet, mining co-occurrence rules from tagging data can be challenging. Firstly, the quality of the rules or the candidate tags is not equally high. Secondly, the number of rules mined can be very large while the applicable rules for each target resource are usually small. To address these challenges, Belem*et al.* [6] proposed to incorporate other quality metrics for scoring both the antecedent and the consequents of the applicable rules. Menezes*et al.* [84] studied *on-demand rule mining* to improve space efficiency in tag prediction. Nonetheless, the rule-based approach requires the target resource to have prior tags, otherwise no rule can be applied for tag prediction. In other words, this approach suffers from cold-start problem when the target resources are untagged.

Classification-based approach trains a binary classifier for each candidate tag, and the target resource is then classified by every candidate classifier to determine the tags to be assigned [47, 115]. Heymann *et al.* [47] applied this approach for predicting tags for Web pages, and examined features such as *page text*, *anchor text* and *the structure of surrounding hosts*. They found that tags can be predicted with high precision but low recall using these three types of features. The main drawback of this approach is that, it is computationally prohibitive. To perform tag prediction for a single target resource, it requires running all the classifiers corresponding to all candidate tags. To reduce the computation overhead, only a small set of tags are selected for building the classifiers, *e.g.* the top 140 frequent tags on Delicious [115]. As a result, this approach largely limits the number of candidate tags that can be predicted.

Cluster-based approach partitions the resources, together with their tags, into clusters. When applied to tag prediction, it first determines the belonging cluster of the target resource, and then scores and ranks the candidate tags within the belonging cluster as predicted tags for the target resource. Song *et al.* [105] proposed to perform clustering by partitioning the bipartite graph formed between resources and their content words. When *hard clustering* is adopted, *i.e.* each resource belongs to one and only one cluster and clusters do not overlap, this approach has the problem of making the same set of predicted tags for all resources that fall into the same cluster.

Table 2.1 summarizes the existing literature based on the above categorization of approaches. In general, the content words of the target resource are required as input for content-based, topic-based, and language-based approaches. Topic-based and language-based approaches model the tag vocabulary separately from the word vocabulary, whereas content-based approach does not. Based on this distinction, we categorize the study by Diaz-Aviles *et al.* [22] into content-based rather than topic-based approach, even though LDA is adopted in their solution. Rule-based approach requires prior tags of the target resource

Table 2.1: Literature for Tag Prediction

Approaches	Assumption for Target Resource				Output Capability		Existing work
	CW	PT	LK	MX	ST	UT	
Content-based	✓				✓	✓	[22, 82, 87]
Topic-based	✓				✓		[18, 52, 61, 71, 102]
Language-based	✓				✓		[31, 108]
Link-based			✓	✓	✓		[5, 17, 72, 97, 107]
Rule-based		✓			✓		[6, 47, 61, 84]
Classification-based				✓	✓		[47, 115]
Cluster-based				✓	✓		[105]

Legend :
CW : content words of the target resource;
PT : prior tags of the target resource;
LK : explicit links between resources, *e.g.* hyperlinks;
MX : may include all of the above and other types of data, *e.g.* domain host;
ST : seen tags, *i.e.* terms in the tag vocabulary;
UT : unseen tags, *i.e.* new terms not yet available in the tag vocabulary.

as input, hence is not applicable to resources that are completely untagged. Link-based approach requires links between resources as input, especially for methods relying on explicit links other than induced links. Classification-based and cluster-based approaches may require the content words and additional features for representing the resources. Lastly, it is worth noting that, content-based approach is the only approach so far that can produce novel tags not yet available in the tag vocabulary. All other approaches model the tag vocabulary explicitly, hence can only predict tags learned before.

2.5 Summary of Topic Model Research

LDA is a probabilistic model for text mining introduced by Blei *et al.* [12]. Given a text corpus with D documents, in which each document has a *bag of words*, LDA assumes that:

- There are K topics in total that describe this corpus;
- Each topic, denoted by k, has a mixture of words;
- Each document, denoted by d, has a mixture of topics;
- Each word token in each document belongs to one of the K topics.

LDA models each topic as a multinomial distribution over the word vocabulary, denoted by ϕ_k, and models each document as a multinomial distribution over the K topics, denoted by θ_d. The K topics are *latent* because they are not directly observed in the corpus. The only variables observed in the corpus are the word tokens for each document. The number of word tokens in document d is denoted by I_d.

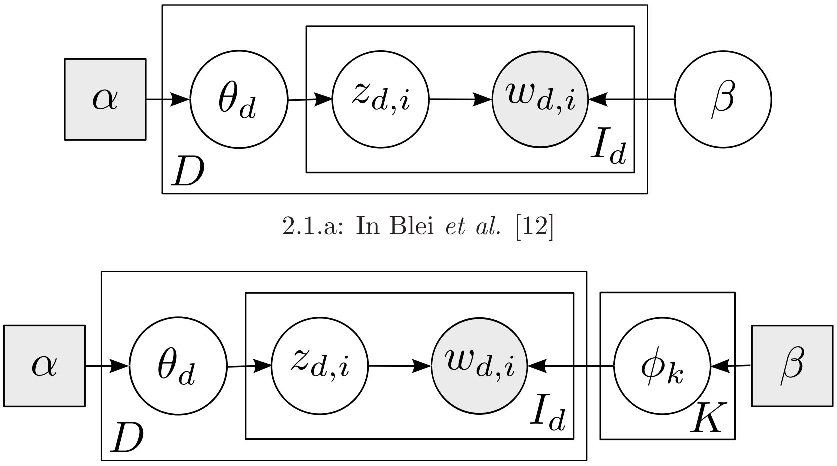
LDA describes a text corpus as the result of a generative process, where individual word tokens, denoted by $w_{d,i}$, are generated by the following two steps:

1. sample a topic $z_{d,i}$ from θ_d of document d, denoted by $z_{d,i} \sim \theta_d$;

2. sample a word $w_{d,i}$ from $\phi_{z_{d,i}}$ of topic $z_{d,i}$, denoted by $w_{d,i} \sim \phi_{z_{d,i}}$.

Where, the symbol $\sim$ denotes the process of *sampling* a variable or distribution from its governing distribution.

Blei *et al.* [12] assumes that the topic mixtures for documents in the corpus are governed by a Dirichlet distribution, denoted by $\theta_d \sim \mathit{Dirichlet}\,(\vec{\alpha})$. The Dirichlet hyperparameter $\vec{\alpha}$ is a vector of dimension K, and the weights in $\vec{\alpha}$ allow priors to be assigned to the K topics. For example, if we have prior beliefs that topic 1 has much higher probabilities of appearing in the corpus than topic 2, then we may assign a much higher weight to topic 1 than to topic 2 to incorporate such beliefs. However, when modeling an unseen text corpus, prior beliefs are unknown, hence symmetric priors are usually adopted, *i.e.* $\alpha_k = \alpha$ for all $k \in [1, K]$ and $\theta_d \sim \mathit{Dirichlet}\,(\alpha)$. In plate notation[4], the original LDA model proposed by Blei *et al.* [12] is shown in Figure 2.1.a.



2.1.a: In Blei *et al.* [12]

2.1.b: In Griffiths and Steyvers [36]

Figure 2.1: Plate Notation for LDA

Legend of plate notation:

circles	:	random variables;
squares	:	hyperparameters;
rectangles	:	*plates*, repeated samples with the number of repetitions noted at the bottom corners;
arrows	:	conditional dependencies between variables;
shaded	:	observed variables;
unshaded	:	latent variables.

[4] A representation of graphical probabilistic models with repeated variables.

It is worth noting that, Blei *et al.* [12] did not assume any Dirichlet distribution governing the word mixtures for topics, *i.e.* ϕ_k. The β in Figure 2.1.a represents a matrix of parameters where the rows correspond to the K topics and the columns correspond to the word vocabulary, *i.e.* $\beta = [\phi_1 \ldots \phi_K]^T$. In a later study, Griffiths and Steyvers [36] modified LDA by making the Dirichlet assumption for these topic multinomials, denoted by $\phi_k \sim Dirichlet\,(\beta)$. Effectively, prior beliefs on the words for topics can be incorporated into the model, similar to α. Moreover, this assumption has also made parameter estimation for this model more efficient by applying a collapsed Gibbs sampling technique. Many other studies that extend LDA follow the model by Griffiths and Steyvers [18, 71, 88], including ours to be introduced in Section 3.2. Figure 2.1.b shows the modified LDA model by Griffiths and Steyvers [36] in plate notation. In this case, β represents a Dirichlet hyperparameter, similar to α.

Following [12], topic modeling has gained intense interest from the research community. While the basic LDA model has been applied for solving various tasks [36, 40, 75], many studies also proposed extended topic models for capturing other characteristics in text corpora [9, 10, 11, 88, 116]. Blei *et al.* [10] proposed a correlated topic model, which captures the hypothesis that, *topics appearing in a document together are likely to appear together in other documents*. In other words, the existence of topics in the same documents are correlated. In another work, Blei *et al.* [11] also proposed a dynamic topic model, which captures the evolution in topic compositions overtime. While the topics may continue to be present in the text corpus, the contributing words for the topics may be different due to topic shifts. This observation is especially noted in corpora of scientific literature. While in these studies, the contributing words to topics consist of only unigrams, Wang *et al.* [116] proposed a topical n-gram model, which captures n-grams for topics. Their basic assumption is that, a word token in a document may be generated either independently from its preceding tokens (as the original LDA) or following the preceding token to make up a n-gram for the particular preceding topic. Each word token in a document belongs to one (and only one) of the K topics. LDA model describes the generative process for each word token in each document by first sampling the topic assignment for the token followed by sampling a word for that topic.

Here, we highlight the correspondence topic model (CorrLDA for short) by Blei *et al.* [9] and the entity-topic model by Newman *et al.* [88]. CorrLDA is proposed for modeling images and their caption words [9]. The model assumes that an image may contain multiple regions, and each word in the image caption corresponds to one of the regions. It follows LDA by modeling image regions as topics, which have multinomial distributions over the word vocabulary. Moreover, it also models regions as distributions of image features. CorrLDA relates the distributions of topics in an image caption with the distributions of regions in the same image by a *correspondence* assumption. Similar idea has also been exploited for modeling named entities and their context words [88]. In a text corpus, named entities are recognized proper nouns representing persons, organizations or locations etc, and context words are those that occur within certain proximity of the named entities. Newman *et al.* [88] modeled topics as multinomials of named entities as well as multinomials of context words, and also related the topics for named entities and the topics for context words by a *correspondence* assumption. For our tag prediction task, we also adopt the *correspondence* assumption to model the coupling between the topics for tags and the topics for content words of Web pages.

2.6 Summary of Tag Recommendation Research

Personalized tag recommendation refers to the task of suggesting tags to a user when she annotates a resource. The user and the resource are known as the *query user* and the *query resource* respectively. There are two settings for the task, namely *batch recommendation* and *dynamic recommendation*.

Batch recommendation : Tags are recommended *once* to the query user and query resource, and these recommendations do not change regardless of any user input during the annotation process. This is the usual setting studied in the literature [57, 76, 95, 118]. Traditional evaluation metrics for recommender systems such as precision and recall are adopted for comparing the performance between recommendation methods.

Dynamic recommendation : Tags are recommended as *a series of recommendations* to query user and query resource, and the recommendations are updated whenever the user picks up or enters a tag of her choice. In other words, the set of tags chosen by the query user are regarded as part of the input at each step. This setting is only seen in [28], where Garg and Weber proposed a novel metric for evaluating the recommendation accuracies in a series of tag recommendations. Intuitively, their metric computes the cost incurred in the annotation process, assuming the user examines the recommended tags sequentially and then enters a tag of her choice either from or outside the set of recommendations.

Compared with the task of tag prediction discussed in the previous section, which focuses on resources not yet tagged or inadequately tagged, the task of tag recommendation does not focus on this kind of resources. Moreover, the query resources often have previously been tagged by other users. One approach to tag recommendation commonly adopted in many social tagging systems is based on *tag frequency* for the query resource. Based on the intuition that, *the tags used by more user may more likely be used again*, this approach selects the top frequent tags for the query resource as recommendations. This selection mechanism works for some users, who follow the general crowd in their choice of tags. For these users, such phenomenon is sometimes referred to as *social influence* or *community influence* [81, 99]. It is worth noting that,

frequency-based approach to tag recommendation does not perform personalization. It recommends the same set of tags to all users given the same query resource.

The challenge of tag recommendation arises from the need for personalization. Since individual users have their own tagging preferences, the tags used by others for the same resource may not suit the given query user. We summarize the existing approaches to personalized tag recommendation into the following five categories.

Collaborative-filtering-based approach applies collaborative filtering techniques to select candidate tags from users who share similar preference with the query user [77]. The recommendation algorithm first selects k-nearest neighbors (kNN for short) for the query user, and then recommends tags that are assigned to the query resource by those neighbors. To compare the preferences between users, each user may be profiled using the set of resources she has annotated or the set of tags she has used. Marinho and Schmidt-Thieme [77] found that using user-tag profile to find kNN neighbors outperformed the user-resource counterpart. They suggested that, a user's tag vocabulary is a better representation of personal preferences. However, the main drawback of the collaborative-filtering-based approach is that, the recommended tags must already exist for the query resource, *i.e.* assigned by other users. This approach may suffer from poor performance if the query user has an exclusive tag vocabulary, *i.e.* tags not yet used by others for the same query resource.

Co-occurrence-based approach estimates the co-occurrence probabilities of two tags given that they are assigned to the same resource. Given the tags previously assigned to the query resource, tags co-occurring with these prior tags are aggregated and ranked for tag recommendation. Previous studies examined various kinds of tag co-occurrences [28, 94, 103, 118]. Rae *et al.* [94] studied the co-occurrence between tags under four types of social context, namely *personal context*, *social contact context*, *social group context* and *collective context* (*i.e.* the entire social tagging community). They computed the co-occurrence probabilities of tags under these contexts, and combined the recommendation candidates from these contexts for recommending tags to users annotating images on Flickr. Wetzker *et al.* [118] examined the co-occurrences between resource tags (*i.e.* tags already assigned to the query resource) and personal tags (*i.e.* tags used by the query user in the past, likely to be preferred by the user). They proposed a *personomy translation* method for personalized tag recommendation. They profiled users by the set of probabilities of translating the resource tags to personal tags, and performed tag recommendation using these probabilities.

Graph-based approach examines the tripartite graph formed among resources, users and tags via the assignment relationships in a folksonomy. Jäschke *et al.* [57] proposed *FolkRank*, a random walk technique which operates on folksonomies. It follows the intuition and formulation of PageRank to compute the stochastic popularity of resources, users and tags. Similar to random walk techniques, Guan *et al.* [38] proposed a heat diffusion algorithm that diffuses heat along links in the multi-type graph from the query resource and the query user. For these two graph-based algorithms, personalization is performed by biasing the preference vector towards the query user and the query resource. Marinho *et al.* [76] described a relational learning approach that recommends tags from the neighborhood in a graph of related objects. In their setting, the objects in the graph are posts in the folksonomy, *i.e.* resource-user pairs. The strength of relations between objects are exploited for estimating the probabilistic weighted average from the neighborhood. However, only simple relations were examined, *i.e.* user-tag profiles.

Gemmell *et al.* [29] examined the effect of ambiguous tags with respect to recommendation performance, where *ambiguous tags* are defined as those appearing in more diverse clusters of resources. They experimented numerous tag recommendation algorithms, including frequency-based recommendation, collaborative filtering, and FolkRank. They found that collaborative filtering and FolkRank were less affected by the presence of ambiguity than other simpler methods, such as those based on tag frequency.

Tensor-based approach applies dimensionality reduction techniques on the 3-dimensional tensor corresponding to the resource-user-tag graph [57, 95, 110]. This approach uses the same data as in graph-based approach, but adopts a multi-dimensional algebraic representation and techniques. Symeonidis *et al.* [110], first unfold the 3-dimensional tensor into three 2-dimensional matrices, apply Singular Value Decomposition (SVD for short) to these 2-dimensional matrices individually, and then combine the decomposed matrices again to derive a denser tensor approximating the original graph. The algorithm then recommends candidate tags whose weights in the dense tensor are above some threshold. Rendle and Schmidt-Thieme [95] introduced two more efficient variants of this approach using canonical decomposition and pairwise interaction tensor factorization.

Topic-based approach leverages topic modeling techniques, *i.e.* LDA, for estimating the probabilities that candidate tags are relevant to the query resource and the query user [63]. Krestel and Fankhauser [63] first formulated the tag recommendation task as a general probabilistic ranking problem: $p(t|r,u) \propto \frac{p(t|r)p(t|u)}{p(t)}$, where r denotes the query resource, u denotes the query user, and t denotes a candidate tag. Krestel and Fankhauser examined a language model and a topic model for estimating $p(t|r)$ and $p(u|r)$. Under the language model, $p(t|r) = \frac{c(t,r)}{\sum_{t'} c(t',r)}$, where $c(t,r)$ denotes the number of times the tag t is assigned to the resource r, and t' denotes any other tag. Under the topic model, $p(t|r) = \sum_k p(t|k)p(k|r)$, where k denotes a latent topic. The probabilities for users, *i.e.* $p(t|u)$, can be formulated correspondingly. They adopted the original LDA for probability estimation.

Note that many of the above approaches do not require the resource content to make tag recommendations. They work well for resource that do not have textual content, *e.g.* Flickr[5] photos. Personalized tag recommendation also adopts an assumption that the query resources should have previously assigned tags by other users. This differentiates the task from tag prediction designed for untagged resources.

2.7 Summary of Emerging Trend Discovery Research

In this section, we first examine the different *emerging trends* studied in the literature. The problem of *discovering emerging trends* refers to different tasks in different contexts, and there has not been a consistent definition. We broadly classify the previous work into *phrase tracking* and *topic tracking* tasks.

Phrase tracking focuses on monitoring the use of phrases as they appear in a stream of documents. In this task, the *emerging trends* are the *phrases* showing interesting or significant changes in use over time. For example, on Twitter[6], the most frequently used phrases in recent time period are considered trending topics[7]. To measure interestingness or significance of the tracked phrases, techniques such as *burst detection* and *trend estimation* can be deployed [35, 66, 85].

Topic tracking focuses on identifying topics and monitoring topic shifts in a stream of documents over a long time span. In this task, the *emerging trends*

[5] `www.flickr.com`

[6] `twitter.com`

[7] `twitter.com/#!/trendingtopics`

are the *changes in the topic structures*. For example, in analyzing scientific publications, one topic tracking task is to find the various research specialties and understand the evolution in these research specialties [10, 11, 12], such as *branching*[8] and *merging*[9] and the *shifts in focus*[10].

Unlike the phrase tracking and topic tracking tasks, which take *phrase-centric* and *topic-centric* approaches respectively, our study on trend discovery takes an *item-centric* approach. We track the publications using their social annotations, including social tags and citing documents. We aim to discover long-term trends revealed by these social annotations. In our study, *emerging trends* are prominent long-term trends associated with various semantic dimensions of the annotated publications.

In what follows, we present an overview of the studies taking topic-centric and phrase-centric (focusing on tags) approaches, and describe techniques for burst detection and trend estimation.

Topic Detection and Tracking

The research on topic detection originates from detecting events from news streams [2]. Traditionally, topic detection is performed by clustering news articles into topics in an *online* fashion: Given the current document in the stream, the task requires it to be assigned to: (i) an existing topic (or event) among a set of previously formed topics (or events); or (ii) a new topic, when no existing topic is found sufficiently similar to the current document. Due to the rising popularity of Weblogs in the last decade, topic detection has also been studied on Weblogs [83], which is one form of user-generated content on the Web. Detecting emerging research frontiers and changes of research focus has also become an active area of research [74]. For detecting topics and topic evolution, the main challenge is to model and relate topics across time.

[8] One topic has developed into multiple subtopics.

[9] Multiple topics merge into one super topic.

[10] The keywords for describing the topics are different from before.

The existing solutions regardless of the content sources can be categorized into *discriminative* and *generative* approaches.

Discriminative approach models each topic as a distribution of words at a certain discrete time windows, and changes in topics are compared using these distributions post hoc [83, 86, 106]. Morinaga and Yamanishi [86] proposed a finite mixture model for representing the topic structures in documents at each time point. Emerging topics are discovered by detecting changes of main component in the finite mixture model. Their detection strategy is based on the theory of *dynamic model selection*. Mei *et al.* [83] proposed to detect subtopic themes and spatiotemporal theme patterns in Weblogs using a probabilistic model. They compared theme life cycles and theme snapshots to observe the evolution of theme patterns. Spiliopoulou *et al.* [106] studied topic detection based on clusters. They proposed a cluster transition model, which monitors changes that involve more than one clusters. Their model is shown effective when evaluated on the ACM Digital Library data set.

Generative approach models the data streams (*e.g.* publications at each publication year) by a generative process, in which topics from an earlier time window impose priors on those in the later time windows [3, 11, 85, 122]. Zhou *et al.* [122] proposed to model topics in scientific literature based on the network relationships between the authors. In their work, topical trends are monitored by counting the number of documents belonging to the topics. They used author information to explain the increasing or decreasing trends of topics from year to year. Blei and Lafferty [11] developed a dynamic topic model, which uses Gaussian processes to relate topics across time. The use of Gaussian processes allows topics at each time window to be centered on topics from the previous time windows, *e.g.* by defining the means. Similar to Zhou *et al.* [122], topic trends are monitored by counting the number of documents belonging to the topics. Gohr *et al.* [32] studied topic adaptation under evolving (possibly infinite) vocabulary. Based on probabilistic latent semantic analysis (PLSA for short) for modeling topics, they proposed a *fold-in* technique that allows topics to be folded in with new words and documents to be folded in with new topics.

Many recent studies following the generative approach to topic evolution adopt LDA to model topics. LDA is originally proposed to model an archive of scientific documents, *i.e.* assuming static topics. AlSumait *et al.* [3] devised an Online-LDA model, which extends LDA to incorporate incremental updates in topics from newly arrived documents. Bolelli *et al.* [14, 15] developed a generative author-topic model (GATM for short) that incorporates the temporal order of documents for detecting topic trends. In GATM, topics discovered at earlier time windows are propagated to later time window(s), via topic priors. They also modeled citation links between publications in addition to the text content to identify the *topic-bearing* words. He *et al.* [44] proposed a citation-aware topic evolution learning model, which incorporates the citation links for modeling topic inheritance between the citing and the cited publications. Their empirical study on a CiteSeer[11] dataset suggested that citations are helpful in tracking the evolution of topics.

Although many studies have taken citation links and citing documents into detecting topics and topic evolution in scientific publications, we are aware of no work on discovering topics and topical trends using social tags. In Chapter 5, we consider both citing documents and social tags to be two forms of social annotations for trend discovery in the annotated publications.

Phrase Tracking

Phrase-centric studies has gained increasing interest as more and more content on the Web are created by individual users instead of mainstream media. Morchen *et al.* [85] proposed to identify emerging biomarkers as they appear in biomedical literature. Leskovec *et al.* [66] monitored quoted phrases in

[11] `citeseer.ist.psu.edu`

news articles and Weblogs. They first clustered phrases that may be mutational variants of the same storyline, and developed quantitative model for quantifying the temporal dynamics of the story lines. Goorha *et al.* [35] proposed to identify *interesting* phrases centered around a named item of interest, *e.g.* a named product or company. Studies on tags are concerned with monitoring and analyzing the *usage frequency*, *popularity* or *meaning* of a single tag [33, 50, 117] or a pair of tags [51] over time.

Hotho *et al.* [50] proposed to compute the *popularities* of tags using an algorithm called FolkRank. FolkRank is formulated similarly to PageRank [91], which simulates random walks on a graph to compute the stochastic stationary probabilities of visiting each node of the graph. In PageRank, all nodes are of the same type, *i.e.* Web pages. Whereas in folksonomies, linked objects may be one of the three types, namely resources, users and tags. Hence, FolkRank first projects the tripartite graph of a folksonomy into a monopartite graph. It then computes the stochastic stationary probabilities for each node in the projected graph. Hotho *et al.* proposed to measure the *popularity change* of a tag by considering the popularity rankings of the tag and the total numbers of tags in the respective time points.

A comparative study is given by Wetzker *et al.* [117], in which they proposed a generative process to model the *usage frequency* of tags. Wetzker *et al.* assumed that, *the frequency of a tag in the current time window follows a binomial process from the previous time window.* They regarded trends as *statistical anomalies*, and monitored tags that have the highest log usage frequency from the estimated model. They also adopted the *popularity change* measure proposed by Hotho *et al.* [50] but measured the frequency rankings of tags instead of the stochastic popularity rankings used in [50]. They found that the popularity change in their measure strongly favors new tags.

More recently, Hsu *et al.* [51] examined the co-occurrence relationships between pairs of tags. By performing regression on the temporal correlations, they identified three types of long-term pair-wise trends, namely *steadily-shifting*, *stabilizing* and *cyclic*. In steadily-shifting trend, the correlation between two tags increases or decreases linearly with time. In stabilizing trend, the correlation between two tags may increase or decrease during the initial period, but as time goes by, it approaches a constant level. In cyclic trend, periodic peaks are found in the correlations between two tags. They demonstrated the usefulness of the estimated trends in a tag prediction task, in which candidate tags are scored by their estimated correlation with the existing tags of the target resource, *i.e.* similar to (co-occurrence) rule-based approach noted in Section 2.4.

Different from the above, Gohr *et al.* [33] proposed to monitor the meanings of tags across time. They observed that, "*tags acquire multiple semantics gradually, as users apply them to disparate documents*" [33]. Based on this observation, they aim to provide summaries using topic prototypes, such that users may efficiently inspect the meanings of a given tag across time without going through all associated documents. They modeled topics using Adaptive-PLSA, which is proposed by them in a precedent work [32], for incorporating the evolving vocabularies in the document streams. They proposed to visualize the summary for each tag using *topic table*, which captures: (i) the top words associated with each topic at each time window; (ii) the similarity between the topic composition at a previous time window and the next; (iii) the relative strength of each topic among all topics at the same time window. They demonstrated the effectiveness of their visual summary by spotting alien words contained in a set of artificially inserted documents.

Time Series Analysis

Trend estimation and burst detection are tasks related to time series analysis. *Burst detection* focuses on identifying the time intervals with unusually large number of messages, mentions of some events, or transactions. It finds transient changes in the arrival of data. In contrast, *trend estimation* finds continuous, relatively long-term trends in data streams.

Traditionally, burst detection monitors an aggregate function defined for time intervals of a fixed size, often called the *sliding window*. It maintains a moving sum of the function values over the sliding windows, and compares the sum with a pre-defined threshold to determine if significant changes are observed. However, this basic approach has its drawbacks. Zhu *et al.* [124] observed that, bursts may be of unknown durations and to monitor many sliding windows of varying sizes simultaneously takes quadratic time, which is inefficient. They proposed a data structure called *shifted wavelet tree* for efficient burst monitoring in elastic time intervals, and a streaming algorithm that detect bursts with time complexity linear to the set of varying sliding window sizes.

One of the most famous algorithm for burst detection is the finite automaton model proposed by Kleinberg [59]. It assumes that, messages arrive in a stream with an arrival rate depending on the underlying state at the moment, and the onset of a *burst* is signaled by a transition from a lower-rate state to a higher-rate state. The advantages of this state automaton model are twofold. First, by assigning cost to the state transitions, one can control the frequency in state changes, preventing very short bursts. Second, "the bursts associated with state transitions form a naturally nested structure, with a long burst of low intensity potentially containing several bursts of higher intensity inside it" [59]. Using this model, Kleinberg demonstrated how a hierarchical structure of burstiness can help to identify "landmark" messages in a large corpus of emails.

More recently, He and Parker [43] challenged the definition of bursts in Kleinberg's model, and proposed to monitor bursts in data streams as the interval in which momentum is increasing. They introduced definitions such as *position*, *mass*, *velocity*, *momentum* and *acceleration* for time series based on kinetics concepts in physics. In the context of detecting bursting MeSH[12] terms (Medical Subject Headings), *position* is a measure of intensity (*e.g.* number of articles containing the term) and *mass* is a measure of importance (*e.g.* number of citations). Using the proposed model, they demonstrated the ability to detect the bursts of MeSH terms, as they became as prominent as to be inserted into the MeSH topic hierarchy. This new technique is shown to outperform Kleinberg's model in identifying bursty intervals.

Trend estimation focuses on long-term trends instead of changes in (relatively) short intervals. Trend estimation on a time series requires constructing a model (*e.g.* function) that can be used to describe the observed data or even to predict future values. Often, the raw measurements of time series data are noisy, *e.g.* stock prices and stock trading volume [54]. To remove short-term fluctuations, *smoothing* may be applied prior to trend estimation, *e.g.* using moving averages [43].

The simplest type of trend is the linear trend, which can be represented as a linear model $f(t) = a \times t + b + \epsilon_t$. In this model, ϵ_t denotes the amount of noise at time t, while a and b are model parameters to be estimated from the time series data. Particularly, parameter a may be used to describe the *upward* and *downward* linear trend. To fit a model to a time series, an optimizer is used to determine the suitable set of parameter values that minimize the error between the model and the actual data values, *e.g.* the sum of square error. Morchen *et al.* [85] deployed such a linear model to describe the emergence of MeSH terms. They monitored the log frequencies of the terms instead of the absolute frequencies. The estimated *increasing* trend parameters were then used to select emerging MeSH terms.

Depending on the data at hand, more complicated models may be needed. Leskovec *et al.* [66] observed that story lines exhibit heart-beat patterns when they are quoted in mainstream and social media. They proposed a model that

[12] `www.ncbi.nlm.nih.gov/mesh`

incorporates *imitation effect* and *recency effect* to mathematically quantify such patterns. Imitation effect takes into account the number of articles written previously on the same story, and recency effect favors new stories over old stories, assuming that all stories are competing for a fixed amount of media attention. Based on the proposed model and a large corpus of news article and blogs, they observed a typical lag of 2.5 hours between the news media and blogs in the peak of attentions to political story lines.

Chapter 3

Tag Prediction: A Topic Modeling Approach

3.1 Introduction

In this chapter, we study how to predict tags for the untagged Web resources. Particularly, we focus on predicting tags for Web pages, which have content words.

Given that tags are keywords that describe or summarize a page, one may propose to select important words from the content as the predicted tags. tf (term frequency) and tf-idf (term frequency × inverse document frequency) are commonly used criteria for selecting important words. However, such selection makes an implicit assumption that tags can only come from words that appear in the page content. In reality, such an assumption does not always hold.

To overcome the limitation of the vocabulary of a single page, one may take the link-based or topic-based approach to predict words or tags from other relevant pages, *e.g.* those sharing similar content. In link-based approach, Web pages are linked via explicit (*e.g.* hyperlinks) or implicit (*e.g.* similarity) links, and tags are harvested from other pages to the target page via these links [5, 17, 72]. In topic-based approach, a collection of web pages may cover multiple topics, and each topic is associated with a set of relevant terms. Tags are predicted by first learning the topics covered by the target page and then estimating the probabilities of candidate tags with respect to the learned topics [18, 22, 102]. As opposed to link-based approach, topic-based approach does not find pair-wise similar pages directly so as to propagate relevant tags. Instead, the relevance of tags are learned via topics, which are learned from many pages.

We observe that tags can be viewed as an abstraction of the content they are assigned to. Often, they are general terms for representing a certain topic. As a single page can cover multiple topics, one would expect the tags, which are collectively assigned by multiple users, to match some of these topics. Based on this observation, we tackle the tag prediction task by leveraging the multi-topic nature of Web pages and the correspondence between tags and topics.

We propose a probabilistic topic model known as LDAtgg to capture these intuitions. LDAtgg models the tag vocabulary explicitly, so that it is capable of representing tags created by the user community of the social tagging system. This overcomes the limitation of selecting content words from the target page alone, such as in the tf and tf-idf approaches. We solve the tag prediction task in two phases, namely training and testing. During training, we learn the probabilities of tags being associated with the topics, based on pages with known tags. During testing, we predict tags for news pages that are untagged.

In this research, we seek to answer the following questions:

(i) What is the relationship between content of Web pages and their tags?

(ii) How can one create a model to incorporate this relationship?

(iii) How can the model be used for the tag prediction task?

(iv) How does the model behave for Web pages with varying characteristics?

Our contribution in this research can be summarized as follows.

- We propose LDAtgg, a probabilistic topic model for solving the tag prediction task. Our model captures the correspondence between tags and topics, as well as the correspondence between content words and topics.

- We formulate a Gibbs sampling procedure for learning the model parameters.

- We evaluate the effectiveness of LDAtgg model and compare it with other tag prediction methods on a real and novel collection of news articles and tagging data from Delicious[1]. Our experimental results show that, topic-based approach outperforms the non-topic-based baselines.

- We conduct in-depth analysis to further examine the strengths and weaknesses of the proposed method. We find that LDAtgg is good at predicting less exclusive tags, *i.e.* tags that are assigned to more pages, but poorer at more exclusive tags. In other words, LDAtgg is good at predicting tags at topical levels, and these tags provide links to Web pages at higher semantic levels. Given this characteristic, we also highlight the possible extensions to the model.

3.2 LDAtgg Model for Tag Prediction

We address the tag prediction task by following topic-based approach, and we focus on predicting tags for Web pages with textual content, such as online news articles. In what follows, we first introduce the notations used in our discussion, and then give our definition to the specific tag prediction task studied in this chapter. Following that, we describe our LDAtgg model for solving the task.

We name each news article (the resources) as a document, and denote it using d. Each document contains a bag of words (extracted from its news

[1] www.delicious.com

content), denoted by $\vec{w}_d$. We use I_d to denote the number of word tokens in $\vec{w}_d$, and use i to denote the sequence number of a word token $w_{d,i}$. In a social tagging system, such as Delicious[2], each annotated document also has a bag of tags, denoted by $\vec{t}_d$, which are collectively assigned by multiple users. We use J_d to denote the number of tag tokens in $\vec{t}_d$, and use j to denote the sequence number of a tag token $t_{d,j}$. Note that, each word token $w_{d,i}$ corresponds to a unique word in the word vocabulary, denoted by w. We use W to denote the size of the word vocabulary. Similar notations are used for a unique tag t and the size of the tag vocabulary T. Table 3.1 lists the symbols we use to describe a corpus of annotated documents, where each document has both words and tags.

Table 3.1: Notations for Data

Symbol	Definition
D	the size of the document collection
W	the size of the word vocabulary
T	the size of the tag vocabulary
d	the index of a document in the collection
w	the index of a word in the word vocabulary
t	the index of a tag in the tag vocabulary
$\vec{w}_d$	the bag of words for document d
$\vec{t}_d$	the bag of tags for document d
I_d	the number of word tokens in document d
J_d	the number of tag tokens in document d
i	the sequence number of a word token in document d
j	the sequence number of a tag token in document d
$w_{d,i}$	the i-th word token in document d
$t_{d,j}$	the j-th tag token in document d

Given the above notations, our approach to the task of tag prediction for textual Web documents can be described as follows: (i) Given a corpus of annotated Web documents, in which each document has a bag of words and a bag of tags, our task is to learn a model, denoted by $\mathcal{M}$, using this set of documents. (ii) When given a new Web document, which contains content words but no social tag is available, our task is to produce a ranked list of candidate tags using the model $\mathcal{M}$.

[2] `delcious.com`

We proposed LDAtgg model to solve the tag prediction task. In LDAtgg, we assume that there are K topics in total that describe this corpus of documents. We use $k \in [1, K]$ to denote an individual topic. Following Blei *et al.* [12], we further assume that each document has a mixture of topics (denoted by θ_d), and each topic has a mixture of words (denoted by ϕ_k) as well as a mixture of tags (denoted by ψ_k). We adopt multinomial distributions to model θ_d, ϕ_k and ψ_k respectively. Note that, we model the tag vocabulary separately from the word vocabulary. In other words, the multinomial distributions ϕ_k and ψ_k are based on two separate vocabularies. The reason is because tags are freeform keywords, which are often not confined to the word vocabulary. For example, some tags are concatenations of phrases such as `socialmedia`, and some tags are very personal to the particular document annotator such as `mustread`.

In LDAtgg, we model the coupling between the topics for tags and the topics for words based on the following assumption, "*if a topic is discussed more often in the content of the document, then it is more likely to have more tags of the same topic assigned to the same document*". This is the *correspondence* assumption adopted in our LDAtgg model. We model such correspondence assumption as a uniform sampling process, in which the topic assignments for tags are sampled *uniformly* from the topic assignments for words. We use $z_{d,i}$ to denote the topic assignment for the word token $w_{d,i}$, and use $y_{d,j}$ to denote the topic assignment for the tag token $t_{d,j}$. Formally,

$$y_{d,j} \sim Uniform\,(z_{d,1}, \ldots, z_{d,I_d}) \qquad \text{(Eq. 3.1)}$$

Given the above assumptions, our LDAtgg model describes the corpus of annotated Web documents as being generated by the following process:

For each document d:

(i) For each word token $w_{d,i} \in \vec{w}_d$:

(a) sample a topic $z_{d,i}$ from the mixture of topics θ_d for d, denoted by

$z_{d,i} \sim \theta_d$;

(b) sample a word $w_{d,i}$ from the mixture of words $\phi_{z_{d,i}}$ for $z_{d,i}$, denoted by $w_{d,i} \sim \phi_{z_{d,i}}$.

(ii) For each tag token $t_{d,j} \in \vec{t_d}$:

(a) sample a topic $y_{d,j}$ *uniformly* from the topic assignments for word tokens in d, denoted by $y_{d,j} \sim Uniform\left(z_{d,1}, \ldots, z_{d,I_d}\right)$;

(b) sample a tag $t_{d,j}$ from the mixture of tags $\psi_{y_{d,j}}$ for $y_{d,j}$, denoted by $t_{d,j} \sim \psi_{y_{d,j}}$.

The entire corpus is generated by repeating the above process for every document. At this point, we would like to note that the above generative process is the mathematical assumptions of the model. It does not mean that the documents must have been produced this way, or the authors must have followed this process to write the content of the documents (or annotators followed this process to assign the tags). It is a mathematical assumption which provides a way for us to explain the words and tags we observe for the documents, when the actual intermediate steps are not available.

Following Blei *et al.* [12], we assume that the mixture of topics for documents are governed by a Dirichlet distribution with prior denoted by α. Following Griffiths and Steyvers [37], we further assume that the mixture of words for topics are governed by a Dirichlet distribution with prior denoted by β, and the mixture of tags for topics are governed by a Dirichlet distribution with prior denoted by γ. Formally,

$$\begin{aligned}\theta_d &\sim Dirichlet\left(\alpha\right)\\ \phi_k &\sim Dirichlet\left(\beta\right)\\ \psi_k &\sim Dirichlet\left(\gamma\right)\end{aligned}$$

Table 3.2 summarizes the symbols we use to describe LDAtgg model. Figure 3.1 depicts LDAtgg model in *plate notation*.

Table 3.2: Notations for LDAtgg Model

Symbol	Definition
K	the number of latent topics
k	the index of a latent topic
$z_{d,i}$	the topic assignment to $w_{d,i}$
$y_{d,j}$	the topic assignment to $t_{d,j}$
θ_d	the multinomial of topics for document d
ϕ_k	the multinomial of words for topic k
ψ_k	the multinomial of tags for topic k
α	the Dirichlet prior for θ_d
β	the Dirichlet prior for ϕ_k
γ	the Dirichlet prior for ψ_k

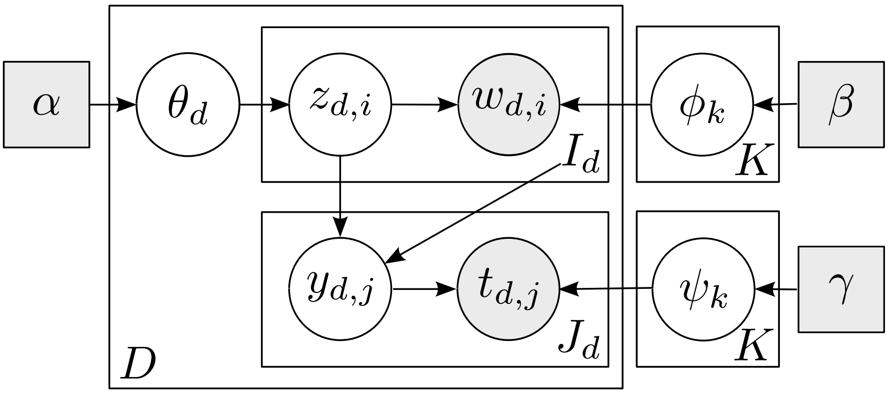


Figure 3.1: Plate Notation for LDAtgg

3.3 Parameter Estimation for LDAtgg Model

In LDAtgg, three sets of model parameters have to be learned to fully describe a corpus of annotated Web document using the model. These parameters are: θ_d, the mixture of topics for documents; ϕ_k, the mixture of words for topics; and ψ_k, the mixture of tags for topics.

We adopt Gibbs sampling, which is a Markov Chain Monte Carlo (MCMC for short) method, to learn the model parameters. Similar to expectation maximization (EM for short) and variational inference techniques, Gibbs sampling performs approximate inference on the model parameters when given the observed variables. It was first adopted by Griffiths and Steyvers [37] for learning topic models, and has been widely adopted as the inference technique for topic models. Its popularity is due to its efficiency in estimating the joint *a posteriori* probability of an individual variable given the assignments of all other variables. Particularly, the joint *a posteriori* probabilities we want to estimate in LDAtgg model are $p\left(z_{d,i}|\vec{z}_{\neg\{d,i\}},\vec{w},\vec{y},\vec{t}\right)$ and $p\left(y_{d,j}|\vec{y}_{\neg\{d,j\}},\vec{t},\vec{z},\vec{w}\right)$. These probabilities are computed by Eq. 3.2 and Eq. 3.3 respectively. We lay out the detailed derivation of these equations in Appendix B. The symbol $\neg\{d,i\}$ denotes the exclusion of the current word token $w_{d,i}$ in the current document.

Algorithm 1 outlines our Gibbs sampler for learning a LDatgg model using training data $\mathcal{D}^{\text{train}}$. During training, the sampler iteratively samples and updates the topic assignments for each word token based on the estimated probabilities $p\left(z_{d,i}|\vec{z}_{\neg\{d,i\}},\vec{w},\vec{y},\vec{t}\right)$, and the *exclusion* denoted by symbol $\neg\{d,i\}$ corresponds to lines 7 in Algorithm 1. Similarly for each tag token, the sampler iteratively computes the probabilities $p\left(y_{d,j}|\vec{y}_{\neg\{d,j\}},\vec{t},\vec{z},\vec{w}\right)$, and the *exclusion* denoted by symbol $\neg\{d,j\}$ corresponds to lines 17 respectively. Table 3.3 lists the symbols used in the sampler.

$$\begin{aligned}
& p\left(z_{d,i}=k|\vec{z}_{\neg\{d,i\}},\vec{w},\vec{y},\vec{t}\right) \\
\propto\ & \frac{n_{d,\neg i}^{k}+\alpha}{\sum_{k'=1}^{K}\left(n_{d}^{k'}+\alpha\right)-1} \\
\times\ & \frac{n_{k,\neg\{d,i\}}^{w_{d,i}}+\beta}{\sum_{w=1}^{V_w}\left(n_{k,\neg\{d,i\}}^{w}+\beta\right)} \\
\times\ & \left(\frac{1+n_{d,\neg i}^{k}}{n_{d,\neg i}^{k}}\right)^{m_d^k} \qquad \text{(Eq. 3.2)}
\end{aligned}$$

$$\begin{aligned}
& p\left(y_{d,j}=k|\vec{y}_{\neg\{d,j\}},\vec{t},\vec{z},\vec{w}\right) \\
\propto\ & \frac{m_{k,\neg\{d,j\}}^{t_{d,j}}+\gamma}{\sum_{t=1}^{V_t}\left(m_{k,\neg\{d,j\}}^{t}+\gamma\right)} \\
\times\ & \frac{n_d^{y_{d,j}}}{I_d} \qquad \text{(Eq. 3.3)}
\end{aligned}$$

A learned model $\mathcal{M}$ can be obtained after we run Algorithm 1 for a sufficient number of iterations, denoted by R. The model parameters, namely θ_d, ϕ_k

Table 3.3: Notations for Gibbs Sampler

Symbol	Definition
$\mathcal{D}^{\text{train}}$	the set of training data, as described in Table 3.1
R	the number of iterations to run the sampler
$\mathbf{N}_D^K$	the parameter matrix of dimension $D \times K$
$\mathbf{N}_K^W$	the parameter matrix of dimension $D \times W$
$\mathbf{N}_K^T$	the parameter matrix of dimension $K \times T$
n_d^k	the entry at the d-th row and the k-th column in $\mathbf{N}_D^K$, which counts the number of word tokens that are assigned to topic k and belong to document d in the training data
n_k^w	the entry at the k-th row and the w-th column in $\mathbf{N}_K^W$, which counts the number of word w that are assigned to topic k in the training data
m_k^t	the entry at the k-th row and the t-th column in $\mathbf{N}_K^W$, which counts the number of tag t that are assigned to topic k in the training data
$\neg\{d, i\}$	exclusion of the i-th word token in document d

and ψ_k, can be computed using the output $\mathbf{N}_D^K$, $\mathbf{N}_K^W$ and $\mathbf{N}_K^T$ according to Eq. 3.4, Eq. 3.5 and Eq. 3.6 respectively. These derivations are based on the expectation of the Dirichlet distribution [45].

$$\theta_d^k = \frac{n_d^k + \alpha}{\vec{n}_d + \vec{\alpha}} = \frac{n_d^k + \alpha}{\sum_{k'=1}^{K} \left(n_d^{k'} + \alpha\right)} \qquad \text{(Eq. 3.4)}$$

$$\phi_k^w = \frac{n_k^w + \beta}{\vec{n}_k + \vec{\beta}} = \frac{n_k^w + \beta}{\sum_{w'=1}^{W} \left(n_k^{w'} + \beta\right)} \qquad \text{(Eq. 3.5)}$$

$$\psi_k^t = \frac{m_k^t + \gamma}{\vec{m}_k + \vec{\gamma}} = \frac{m_k^t + \gamma}{\sum_{t'=1}^{T} \left(m_k^{t'} + \gamma\right)} \qquad \text{(Eq. 3.6)}$$

One run of the Gibbs sampler is called one *chain*. Each iteration of the chain produces a *sample* of the model. Since Gibbs sampling is an MCMC process, the first few samples are often regarded as *burn-in*. For stable estimation, we average the model parameters from multiple samples of the same chain after burn-in, as suggested in [37, 45]. For tag prediction, we also average the probabilities estimated for candidate tags from multiple chains, as we will discuss in Section 3.4.2.

In the prediction phase, given each test document d', where only the content words ($\vec{w}_{d'}$) are observed, we are to produce a ranked list of candidate

Algorithm 1: Gibbs Sampler for LDAtgg - Training

```
Input: $\mathcal{D}^{\text{train}}$, $R$, $K$, $\alpha$, $\beta$, $\gamma$
Output: $\mathbf{N}_D^K$, $\mathbf{N}_K^W$, $\mathbf{M}_K^T$
initialize $z_{d,i}$ and $y_{d,j}$ randomly;
initialize $n_d^k$, $n_k^w$, and $m_k^t$ according to $z_{d,i}$ and $y_{d,j}$;
repeat
    for $d = 1$ to $D$ do
        for $i = 1$ to $I_d$ do
            remove current assignment $z_{d,i}$ of $w_{d,i}$;
            decrement $n_{z_{d,i}}^{w_{d,i}}$ and $n_d^{z_{d,i}}$;
            for $k = 1$ to $K$ do
                compute $p\left(z_{d,i} = k|\vec{z}_{\neg\{d,i\}}, \vec{w}, \vec{y}, \vec{t}\right)$ according to Eq. 3.2;
            end
            sample a topic $z$ from $p\left(\vec{k}|\vec{z}_{\neg\{d,i\}}, \vec{w}, \vec{y}, \vec{t}\right)$;
            $z_{d,i} \leftarrow z$;
            update $n_z^{w_{d,i}}$ and $n_d^z$;
        end
        for $j = 1$ to $J_d$ do
            remove current assignment $y_{d,j}$ of $t_{d,j}$;
            decrement $m_{y_{d,j}}^{t_{d,j}}$;
            for $k = 1$ to $K$ do
                compute $p\left(y_{d,j} = k|\vec{y}_{\neg\{d,j\}}, \vec{t}, \vec{z}, \vec{w}\right)$ according to Eq. 3.3;
            end
            sample a topic $y$ from $p\left(\vec{k}|\vec{y}_{\neg\{d,j\}}, \vec{t}, \vec{z}, \vec{w}\right)$;
            $y_{d,j} \leftarrow y$;
            update $m_y^{t_{d,j}}$;
        end
    end
until $R$ iterations ;
```

tags based on the learned model $\mathcal{M}$. In other words, we estimate the probability $p\left(t|\vec{w}_{d'}; \mathcal{M}\right)$ for each candidate tag seen in the training data, and rank these candidate tags by their probabilities. In LDAtgg model, the estimated probability for a candidate tag t is computed by Eq. 3.7.

$$p\left(t|\vec{w}_{d'}; \mathcal{M}\right) = \textstyle\sum_{k=1}^{K} p\left(t|k; \mathcal{M}\right) p\left(k|\vec{w}_{d'}; \mathcal{M}\right) \qquad \text{(Eq. 3.7)}$$

To compute $p\left(k|\vec{w}_{d'}; \mathcal{M}\right)$, we re-sample the topic assignments for the word tokens in d', using the model parameter ϕ_k learned from training data. Different from sampling in the training phase, re-sampling for test documents does not involve the steps for tag tokens. Algorithm 2 outlines the Gibbs sampler for re-sampling for test documents. The joint *a posteriori* probability for each word token at each step is derived according to Eq. 3.8. Note that, we use the symbol $\tilde{n}$ to denote the counts in the test documents, differentiating from n in the learned model.

Algorithm 2: Gibbs Sampler for LDAtgg - Prediction

```
Input: $\mathcal{D}^{\text{test}}$, $R'$, $K$, $\alpha$, $\beta$, $\mathbf{N}_K^W$
Output: $\tilde{\mathbf{N}}_{D'}^K$, $\tilde{\mathbf{N}}_K^W$
initialize $z_{d',i}$ and $y_{d',j}$ randomly;
initialize $\tilde{n}_{d'}^k$, $\tilde{n}_k^w$ according to $z_{d',i}$ and $y_{d',j}$;
repeat
    for $d' = 1$ to $D'$ do
        for $i = 1$ to $I_{d'}$ do
            remove current assignment $z_{d',i}$ of $w_{d',i}$;
            decrement $\tilde{n}_{z_{d',i}}^{w_{d',i}}$ and $\tilde{n}_{d'}^{z_{d',i}}$;
            for $k = 1$ to $K$ do
                compute $p\left(z_{d',i} = k|\vec{z}_{\neg\{d',i\}}, \vec{w}\right)$ according to Eq. 3.8;
            end
            sample a topic $z$ from $p\left(\vec{k}|\vec{z}_{\neg\{d',i\}}, \vec{w}\right)$;
            $z_{d',i} \leftarrow z$;
            update $\tilde{n}_z^{w_{d',i}}$ and $\tilde{n}_{d'}^z$;
        end
    end
until $R'$ iterations ;
```

$$
\begin{aligned}
& p\left(z_{d',i} = k|\vec{z}_{\neg\{d',i\}}, \vec{w}; \mathcal{M}\right) \\
\propto\ & \frac{\tilde{n}_{d',\neg i}^{k} + \alpha}{\sum_{k'=1}^{K}\left(\tilde{n}_{d'}^{k'} + \alpha\right) - 1} \\
\times\ & \frac{n_k^{w_{d',i}} + \tilde{n}_{k,\neg\{d',i\}}^{w_{d',i}} + \beta}{\sum_{w=1}^{V_w}\left(n_k^w + \tilde{n}_{k,\neg\{d',i\}}^{w} + \beta\right)}
\end{aligned}
\tag{Eq. 3.8}
$$

Time Complexity

For time complexity analysis for Algorithms 1 and 2, let us count the number of times Equations Eq. 3.2 and Eq. 3.3 are executed during training, and the number of times Equation Eq. 3.8 is executed during re-sampling for test documents.

For training, Eq. 3.2 is executed for each latent topics (K topics) for each word token (I_d word tokens) in each training document (D documents) during each iteration (R iterations), and Eq. 3.3 is executed for each latent topics for each tag token (J_d tag tokens) in each training document during each iteration. Let us use I to denote the maximum number of word tokens in a document, and J to denote the maximum number of tag tokens in a document. Then, the upper bound time complexity for Gibbs sampler for training is $O\left(KD\left(I+J\right)R\right)$.

For re-sampling for test document, Eq. 3.8 is executed for each latent topic (K topics) for each word token ($I_{d'}$ word tokens) in each test document (D' documents) during each iteration (R' iterations). Hence, the upper bound time complexity for Gibbs sampler for re-sampling is $O\left(KD'IR'\right)$.

In summary, the proposed Gibbs sampler is a linear algorithm with respect to all the variables K, D, $I+J$ and R.

3.4 Dataset and Experimental Settings

We conduct experiments to evaluate the proposed LDAtgg model for the tag prediction task. In this section, we first describe a novel dataset we collected from the Web, and then report our measurements and analysis on the model performances using this dataset. We include as evaluation baselines an extensive set of content-based and topic-based methods, noted in Chapter 2. Lastly, we report the run time of LDAtgg using our Gibbs sampler.

3.4.1 Data Preparation

We study tag prediction for online news articles. We chose this type of Web documents because of the availability of text content. We collected news articles from three online publishers, namely BBC[3], CNN[4] and USAToday[5]. We notice that, all these publishers support social tagging via links to Delicious[6] and other social media sharing tools. Our objective in data crawling is to collect as many as possible news articles that contain both text content and tags.

Starting from the home page of each news publisher, we performed breath first search by following hyperlinks that are confined to the respective domain. We searched up to a maximum depth of 4 to obtain an initial set of URLs. For each URL in the initial set, we crawled the HTML source to extract news content. We extracted news content by removing HTML markups and advertisements. Meanwhile, we acquired tags from Delicious for URLs in this initial set. We noted that not all news pages contain text content, and not all have attracted tagging on Delicious. Our final set of URLs for each publisher is the intersection of the set of URLs that contain text and the set of URLs that have been assigned tags in Delicious. In other words, every URL in the final set has text content as well as tags. Our crawls of the news content and tags were conducted in April 2009. A summary of statistics about our data crawling is shown in Table 3.4.

Table 3.4: Statistics for URLs in Data Crawling

	Initial set	Contains text		Attracted bookmarks		Final set
BBC	6,887	4,836	(70.22%)	2,352	(34.15%)	1,956
CNN	7,894	5,702	(72.23%)	2,330	(29.52%)	1,994
USAToday	7,088	3,067	(43.27%)	544	(7.67%)	543
					Total URLs	4,493

As shown in Table 3.4, news pages that have attracted bookmarks only constitute a small portion among those in the initial set, which varies from 7.68% to 34.15% for the three news publishers. This demonstrates that scarcity

[3] news.bbc.co.uk

[4] www.cnn.com

[5] www.usatoday.com

[6] www.delicious.com

in tags is prevalent.

Dataset Statistics

The dataset used in our experiments consists of the union of the final sets of URLs for the three news publishers. The dataset contains 4,493 URLs or *documents* hereafter. We preprocessed the documents by tokenizing them into words, normalized all words to lowercase, and removed stopwords[7]. We further removed words that appear in fewer than 3 documents. The preprocessed vocabulary contains 24,322 words.

We collected 33,222 bookmarks for these URLs made by 16,272 users from Delicious. Delicious tokenizes tags using whitespace by default. We further preprocessed tags by removing the prefixing and suffixing punctuations and normalized them to lowercase. This has resulted in a vocabulary of 12,468 tags.

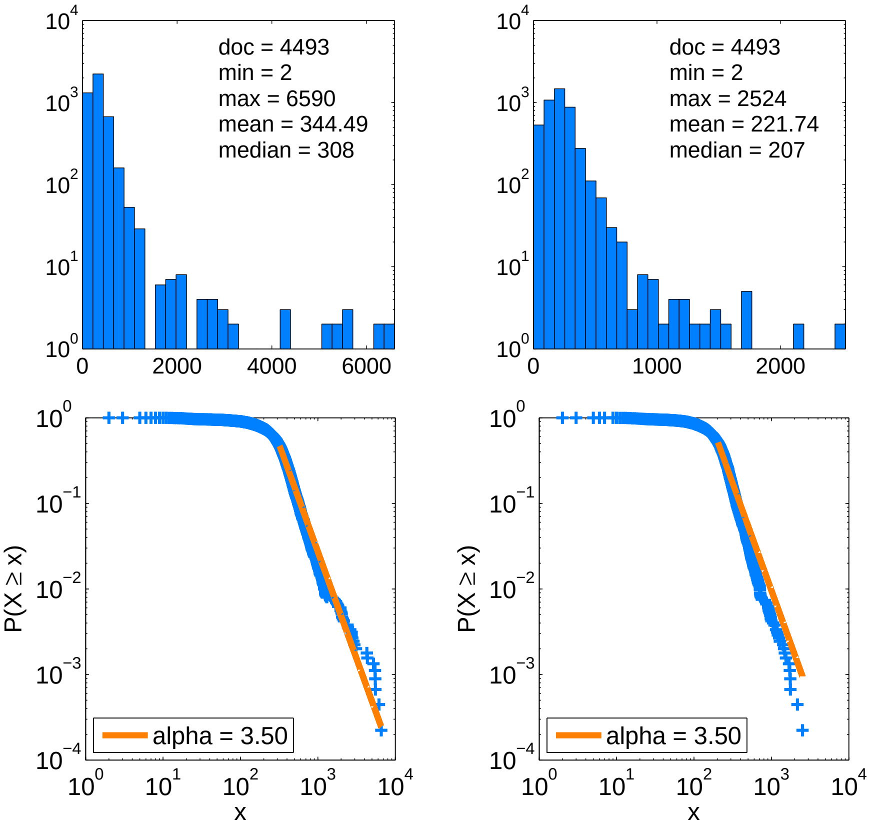
Table 3.5: Statistics for the Dataset

Number of documents	4,493
Number of users	16,272
Number of bookmarks	33,222
Size of word vocabulary	24,322
Size of tag vocabulary	12,468
Average number of word tokens per document	344.49
Average number of tag tokens per document	16.64

Table 3.5 summarizes the statistics for documents in this dataset. We did not stem words since it is the common practice when learning LDA models on text corpora [12, 37]. We did not stem tags as well since tag predictions are commonly evaluated by comparing exact match of tag terms instead of their stems. As expected, the word vocabulary and the tag vocabulary are not identical, as shown in Table 3.5.

After preprocessing, we obtained the bag of words and bag of tags representations for each documents in the dataset. We now analyze the word distribution and tag distribution in documents. Figure 3.2 shows the histograms

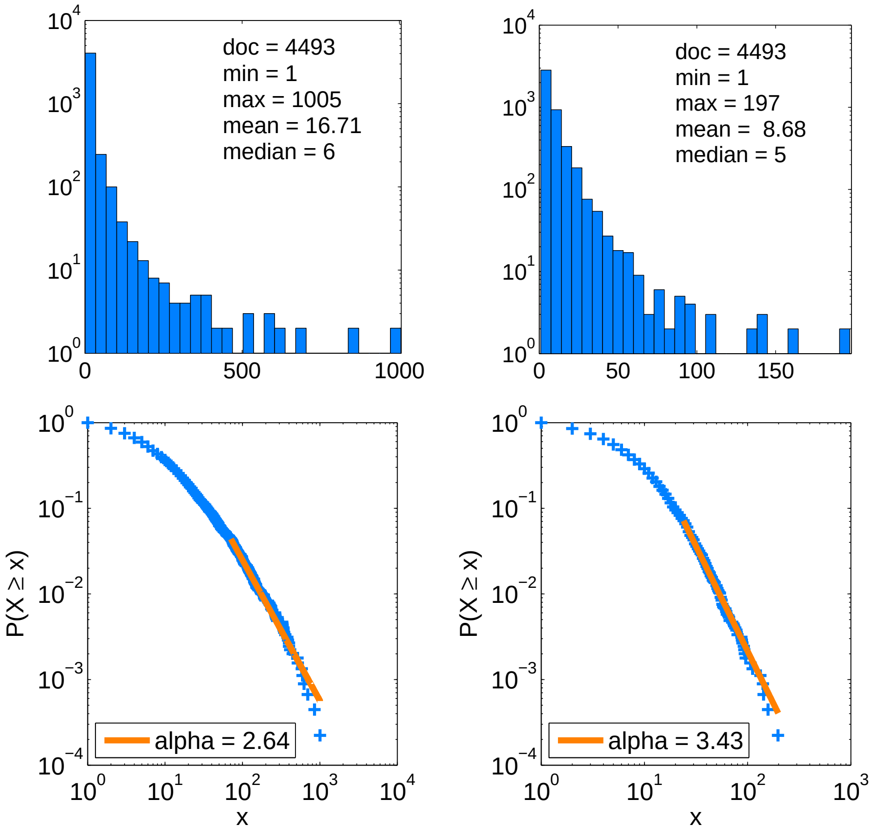
[7] `http://www.textfixer.com/resources/common-english-words.txt`



3.2.a: Word Tokens in Documents 3.2.b: Distinct Words in Documents

Figure 3.2: Distributions for Words in Documents

(upper row) and the cumulative probability densities (lower row) of words for documents, and Figure 3.3 shows the same statistics for tags. We observe that the number of words and tags for documents are not equal. The longest document contains 6,590 word tokens, with 2,524 distinct words. The document mostly bookmarked contains 1,005 tag tokens, with 197 distinct tags. However, as the median values suggest, many documents contain fewer than 308 word tokens and 207 distinct words, and are assigned fewer than 6 tag tokens and 5 distinct tags. The distributions for tags (Figures 3.3.a and 3.3.b) are more stewed than those for words (Figures 3.2.a and 3.2.b respectively). Figures 3.3.a and 3.3.b also show that the cumulative probability densities of tags demonstrate power law distributions, as many social tagging activities do [34, 96, 119]. However, the cumulative probability densities of words in documents are less power-law-like than those for tags, for there are clear bends between the long heads and the power law fits in Figure 3.2.a and Figure 3.2.b.



3.3.a: Tag Tokens in Documents

3.3.b: Distinct Tags in Documents

Figure 3.3: Distributions for Tags in Documents

To better understand the documents in this dataset, we also summarize the frequency of words and tags. In Tables 3.6, we list the top 10 words and top 10 tags with the highest document frequency (shown in the column *Doc*) and their corresponding cumulated frequency (shown in the column *Cum*), *i.e.* the sum of frequencies in all documents.

Table 3.6: Statistics for Frequent Words and Tags

Word	Frequency	
	Doc	Cum
people	2,759	9,787
time	2,447	5,649
years	2,270	5,058
year	1,954	4,310
make	1,928	3,928
world	1,707	4,251
back	1,593	2,916
work	1,569	3,431
made	1,563	2,535
cnn	1,562	2,960

Tag	Frequency	
	Doc	Cum
news	821	2,063
bbc	587	1,564
politics	573	1,574
cnn	465	7,94
health	397	1,521
articles	255	354
obama	248	836
article	246	418
science	234	841
technology	224	859

Table 3.6 suggests that, the document frequencies for the top 10 words are high (*i.e.* from more than 1/3 to 1/2 of the documents in the dataset), whereas the document frequencies for the top 10 tags are much lower (*i.e.* from around 1/20 to 1/5 of the documents in the dataset). It is interesting to note that, the top words are seldom the top tags, except the word `cnn`. Moreover, the top words do not seem to represent clear topics. In contrast, 4 of the top 10 tags represent clear topics, namely `politics`, `health`, `science` and `technology`. The high frequencies of tags such as `news` and `articles` (or `article`) are not a surprise, since our Web documents are all news articles. The same can be said for the domain names `bbc` and `cnn`.

3.4.2 Experimental Setup

Parameter Settings

We evaluate our `LDAtgg` model for the tag prediction task based on 5-fold cross-validation. We split all documents in the dataset into five equal-sized folds, such that when taking one fold as the testing set, the other four folds are regarded as the training set. Table 3.7 shows the statistics for these five folds.

Table 3.7: Statistics for the Five Folds

Fold	Training			Testing		
	D	I	J	D	I	J
1	3,595	1,224,287	59,237	898	323,508	15,821
2	3,594	1,241,733	57,945	899	306,062	17,113
3	3,594	1,249,681	61,514	899	298,114	13,544
4	3,594	1,234,732	60,545	899	313,063	14,513
5	3,595	1,240,747	60,991	898	307,048	14,067

Legend:
D : the total number of documents;
I : the total number of word tokens in all documents;
J : the total number of tag tokens in all documents.

For training, we learned the model parameters ϕ_k and ψ_k from 3 independent chains. Each chain was seeded randomly. We run the Gibbs sampler for 1,000 iterations, where the first 100 iterations were for burn-in and the succeeding samples at every 100 iterations were collected for model estimation. Random seeds were re-drawn at every 100 iterations. For re-sampling on test documents, we run the Gibbs sampler for 20 iterations. These settings are suggested by Heinrich [45] and are commonly practiced for topic models.

We trained our LDAtgg model with predefined 50 topics and 100 topics for the tag prediction task. We set the Dirichlet hyperparameters as $\alpha = \frac{1}{K}$, $\beta = 0.01$ and $\gamma = 0.01$ as suggested by Griffiths and Steyvers [36]. These values were fixed for all folds and all chains.

Evaluation Metrics

We adopt five evaluation metrics, namely precision, recall, f1, r-precision and NDCG. For evaluation metrics other than r-precision, we examine the top 5 predictions for each test document, and report the performance for each method averaged over all test documents. The choice of top 5 follows the convention in previous studies [56].

We define the evaluation metrics for a test document d' as follow. Let t_i denote the predicted tag ranked at position i by a method, $n_{d'}$ denote the number of distinct true tags assigned to the particular test document d', and p denote the current position in the ranked list at which evaluation takes place, *i.e.* $p \in [1, 5]$. Let function $I(d', t_i)$ return 1 if t_i matches any one of true tags assigned to d' and 0 otherwise. We define precision@p, recall@p and f1@p in Eq. 3.9, Eq. 3.10 and Eq. 3.11 respectively.

$$\text{precision@p} = \frac{\sum_{i=1}^{p} I(d', t_i)}{p} \qquad \text{(Eq. 3.9)}$$

$$\text{recall@p} = \frac{\sum_{i=1}^{p} I(d', t_i)}{n_{d'}} \qquad \text{(Eq. 3.10)}$$

$$\text{f1@p} = \frac{2 \times \text{precision@p} \times \text{recall@p}}{\text{precision@p} + \text{recall@p}} \qquad \text{(Eq. 3.11)}$$

As noted in Figure 3.3, more than half of the documents in our dataset have fewer than 5 ground truth tags. This may due to that, at the time of data collection, the news articles have not attracted much attention or interest from Delicious users. To alleviate the adverse effect of tag scarcity on the evaluations, we also measure r-precision. R-precision is defined as the precision@$n_{d'}$, where $n_{d'}$ is the number of distinct true tags for the test document d'. At $p = n_{d'}$, precision@p=recall@p.

While precision, recall, f1 and r-precision are based on binary judgement of relevance, *i.e.* $I(d', t_i) = 1$ or 0, NDCG (Normalized Discounted Cumulative Gain) evaluates prediction accuracy when tags are associated with multiple levels of relevance [55]. DCG@p is defined as

$$\text{DCG@p} \quad = \quad \textstyle\sum_{i=1}^{p} \frac{2^{s(i)}-1}{\log(i+1)} \qquad \text{(Eq. 3.12)}$$

where $s(i)$ denotes the relevance level of the predicted tag at position i, *i.e.* t_i. NDCG@p is DCG@p normalized by the optimal DCG@p for the particular test document, *i.e.* tags with higher levels of relevance always precede those with lower levels of relevance. For the choice of $s(i)$, there has not been a standard scoring methods for the task of tag prediction. However, the frequencies observed for the ground truth tags of each test documents provide a reasonable reference. Au Yeung *et al.* [5] used the tag frequency directly in their NDCG evaluation. However, they also noted that, the resulting NDCG@p performances stayed almost constant across different values of p. They attributed such effect to the skewed distributions of the tag frequencies for documents.

In our study, we address such skewness by mapping the absolute frequencies of the ground truth tags into discrete relevance levels of a small range, *e.g.* $[0, 2]$. It is noted that, as more and more users annotate the same resource, the top tags for the resource become stabilized [34, 96]. In other words, even though the absolute frequencies for the most frequently assigned tags to a document increase over time, the frequency ranks of these tags do not change much. Hence, our mapping should assign higher relevance levels to tags of higher frequency, while at the same time, the discretized relevance levels should not change drastically with respect to the number of annotating users of individual document. Based on this intuition, we experimented with six mapping variations for deriving the relevance level ($s(i)$) from the observed tag frequency ($f_{d'}^{t}$). However, the six variations demonstrate the same qualitative results. Here we report only one representative of the mappings. We call this representative map2, and the derivation from $f_{d'}^{t}$ to $s(i)$ in map2 is shown in Table 3.8. $P_{d'}^{x}$ denotes the x-percentile in the observed tag frequencies for d'.

Table 3.8: Scoring for Ground Truth Tags in NDCG Evaluation

Notation	$s(i)$	Derivation
map2	0	$f_{d'}^{t} = 0$
	1	$f_{d'}^{t} \in [1, P_{d'}^{50})$
	2	$f_{d'}^{t} \in [P_{d'}^{50}, P_{d'}^{100})$

Methods Compared

We include tag prediction methods following content-based approach and topic-based approach for performance comparison. The **content-based** methods include:

tf This method selects keywords from content of the particular test document as tag predictions. The keywords are scored and ranked by their term frequencies in the particular test document. It follows the intuition that, *the keywords appearing in the document more often, the more likely they are used for annotating the document.* Formally, $p(t = w|d') \propto f_{d'}^{w}$. Note that this method does not rely on any training data.

tf-idf This method follows a similar procedure as tf for selecting keywords from document content. It differs only in the way the keywords are scored and ranked by the product of term frequency and inverse document frequency in the dataset. The inverse document frequency gives higher importance to more exclusive keywords in the test document. Formally,

$p\left(t=w|d'\right) \propto f_{d'}^{w} \times \log \frac{D}{\sum_d I(d,w)}$, where D denote the total number of documents in the entire dataset, and $I\left(d,w\right)$ is an indicator function that returns 1 if keyword w appears in the content of the document denoted by d, and 0 otherwise.

LDA This method also selects content words as tag predictions, but these words are not directly extracted from the particular test document. It adopts LDA for learning a topic model from the content words of a collection of training documents, where each topic is parameterized by the posterior probabilities $p\left(w|k\right)$. To predict tags for a test document d', it first estimates the likelihoods that each topic is covered by the document, denoted by $p\left(k|d'\right)$, and then computes the likelihood of a candidate keyword w from the topic posteriors. Formally, $p\left(t=w|d'\right) = \sum_k p\left(w|k\right) \times p\left(k|d'\right)$.

The **topic-based** methods for comparison include our proposed **LDAtgg** model, and the **tagLDA** model proposed by Si and Sun [102]:

tagLDA Figure 3.4 shows **tagLDA** model in plate notation. This model differs from LDAtgg model only in modeling the topic variable $y_{d,j}$ for the tag token $t_{d,j}$. Specifically, LDAtgg assumes that the distribution of $\vec{y_d}$ follows the same distribution of $\vec{z_d}$ for the same document d, which is realized by sampling $y_{d,j}$ uniformly from $\vec{z_d}$. In contrast, tagLDA assumes that $\vec{y_d}$ and $\vec{z_d}$ are both independently drawn from θ_d, meaning that there is no direct correspondence between $\vec{y_d}$ and $\vec{z_d}$. For probability estimation in the tagLDA model, we follow the formulations given by Si and Sun [102]. Following the same notations in Tables 3.2 and 3.3, Eq. 3.13 and Eq. 3.14

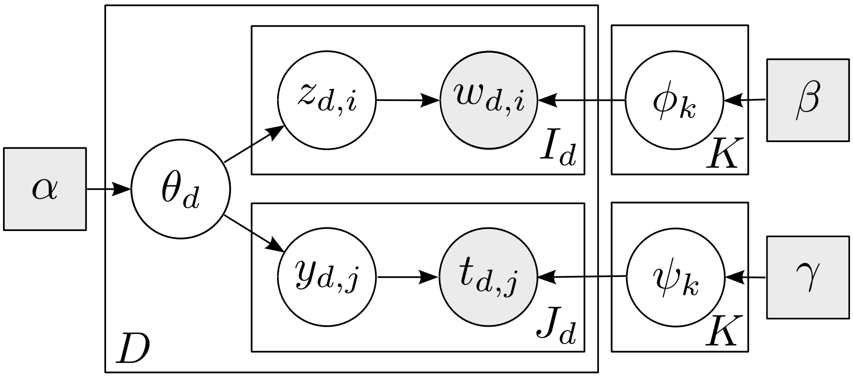


Figure 3.4: Plate Notation for tagLDA

estimate the joint probabilities in tagLDA.

$$\begin{aligned}
& p\left(z_{d,i}=k|\vec{z}_{\neg\{d,i\}},\vec{w},\vec{y},\vec{t}\right) \\
\propto \quad & \frac{n^{k}_{d,\neg i}+\alpha}{\sum_{k'=1}^{K}\left(n^{k'}_{d}+\alpha\right)-1} \\
\times \quad & \frac{n^{w_{d,i}}_{k,\neg\{d,i\}}+\beta}{\sum_{w=1}^{V_w}\left(n^{w}_{k,\neg\{d,i\}}+\beta\right)+\sum_{t=1}^{V_t}\left(m^{t}_{k}+\gamma\right)}
\end{aligned} \tag{Eq. 3.13}$$

$$\begin{aligned}
& p\left(y_{d,j}=k|\vec{y}_{\neg\{d,j\}},\vec{t},\vec{z},\vec{w}\right) \\
\propto \quad & \frac{n^{k}_{d,\neg i}+\alpha}{\sum_{k'=1}^{K}\left(n^{k'}_{d}+\alpha\right)-1} \\
\times \quad & \frac{n^{t_{d,j}}_{k,\neg\{d,j\}}+\gamma}{\sum_{t=1}^{V_t}\left(m^{t}_{k,\neg\{d,j\}}+\gamma\right)+\sum_{w=1}^{V_w}\left(n^{w}_{k}+\beta\right)}
\end{aligned} \tag{Eq. 3.14}$$

All methods that require topic modeling on the dataset, namely LDA, LDAtgg, and tagLDA, follow the same experimental settings, such as the partitions for cross-validation, the number of latent topics, the Dirichlet hyperparameters, and the number of chains and samples for model estimations.

It is worth noting that the aforementioned methods following content-based and topic-based approaches have been studied previously. For content-based methods, while LDA can be compared to that proposed by Diaz-Aviles *et al.* [22], and both tf and idf metrics are covered in [82], though Medelyan *et al.* [82] extracted n-grams (where $n \leq 3$) instead of only unigrams and examined other features based on a Wikipedia[8] corpus. For topic-based methods, the main dis-

[8] www.wikipedia.org

tinction between LDAtgg and tagLDA is in the coupling between the topics for tag tokens and the topics for word tokens: LDAtgg makes the *correspondence* assumption, whereas tagLDA makes the *conditional independent* assumption[9]. Lu *et al.* [71] introduced an additional set of latent variables for modeling the users' perspectives in annotating resources. They also adopted the *correspondence* assumption in coupling the topics for tag tokens and the topics for word tokens. Their study attempted to model two kinds of tags, namely tags that are likely generated by the resource topics (*i.e.* topical and factual tags) and tags that are likely generated by individual users who annotate the resource (*i.e.* subjective and personal tags). One direct application of their model to tag prediction is to use topical tags exclusively, other than user tags. However, they did not explore this direction in [71], but suggested as future research.

3.5 Experimental Results

We first compare the tag prediction accuracy of the above methods using evaluation metrics defined and conduct significance test (Section 3.5.1). To understand the prediction performances between content-based methods and topic-based methods, we further analyze the prediction results with respect to three characteristics of tags and documents, namely *obviousness* of the tags (Section 3.5.2), *adequacy* and *exclusiveness* of the ground truth tags for documents (Section 3.5.3). Following that, we identify two types of false positive errors commonly found in the tag prediction methods, namely *morphological variations* and *partial matches* (Section 3.5.4). To understand the prediction performances between LDAtgg and tagLDA, we identify and examine example documents for which the two methods give contrasting prediction results (Section 3.5.5). Lastly, we report the run time of these topic-based methods using Gibbs sampler (Section 3.5.6).

[9] In other studies, such as [18] and [71], topic models formulated similarly to LDAtgg is also referred to as CorrLDA, and topic models formulated similarly to tagLDA is also referred to as CI-LDA.

3.5.1 Prediction Accuracy

Precision, Recall and F1

Figure 3.5 shows the precision@p, recall@p and f1@p measurements for all methods at the top 5 predictions, *i.e.* $p \in [1, 5]$. These measurements are the micro-averages derived from all test documents in the five folds. On the whole, Figure 3.5 suggests that topic-based methods outperform content-based methods.

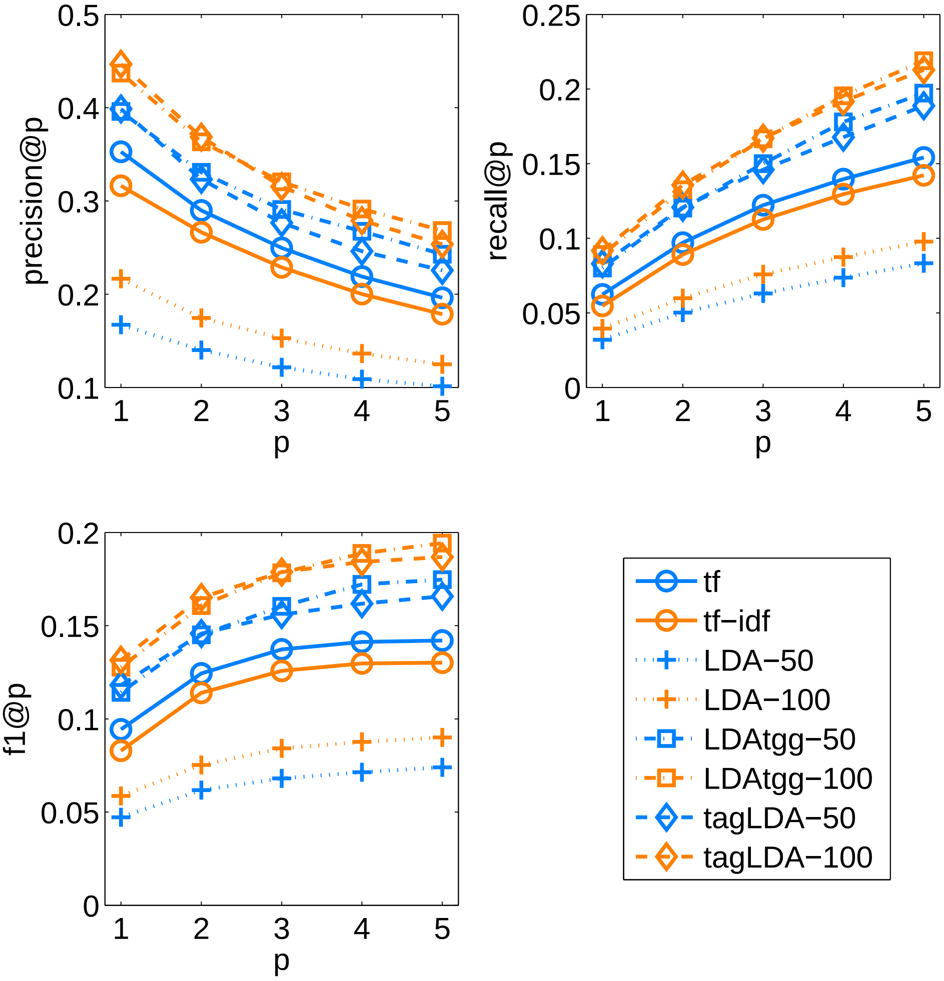


Figure 3.5: Tag Prediction Accuracy

Among the content-based methods, tf and tf-idf, which extract content words from the target document, outperforms LDA, which selects top words from the estimated topics. This observation suggests that top words selected from the estimated topics are generally less relevant to the target documents than frequent words in the document content, even though the topics are relevant. One possible reason for this is that the corpus for topic learning in each fold covers many diverse documents other than the target document itself. This result is different from the work by Diaz-Aviles *et al.* [22], which applied LDA for selecting top content words from topics as tag predictions. Diaz-Aviles *et al.* first constructed an ad hoc corpus for each target document by querying the Web, such that the ad hoc corpus was centered around the target document. In their setting, LDA gives strong performance in tag prediction. Nonetheless, they did not provide comparison with methods that extract content words from the target document itself, such as tf and tf-idf.

We also note that, tf gives more accurate tag predictions than tf-idf. It suggests that, users are likely to select content words that appear frequently in the document as tags, but rare content words are less preferred.

Among the topic-based methods, tagLDA yields better accuracy than LDAtgg at small p, *i.e.* $p = 1$ and $p = 2$, but LDAtgg is superior to tagLDA at larger p, especially when $p \geq 4$. This holds for precision, recall, as well as f1 measurements. It is not a surprise that methods based on these two topic models give competitive performances. In Section 3.5.5, we compare their prediction results in greater detail.

Between the two settings of K, *i.e.* the number of topics assumed for the corpus, LDAtgg-100 always outperforms LDAtgg-50, and tagLDA-100 always performs better than tagLDA-50. This observation suggests that finer-grained topics produces higher tag prediction accuracy for topic-based methods. Hence, we may expect improved tag prediction accuracy using even larger number of topics. However, the higher K, the more complex the model becomes, and the longer it takes the model to learn. Moreover, given that the amount of training data remains unchanged, the higher K, the more sparse data are available for learning each topic. This may lead to *overfitting*. For this dataset, we find using both $K = 50$ and $k = 100$ give reasonably good tag prediction performance.

NDCG

Figure 3.6 shows the NDCG@p measurements for all methods. As noted in Section 3.4.2, we apply map2 to derive the relevance level $s(i)$ for each ground truth tag of the test document using its observed frequency $f_{d'}^{t}$. For comparison, we also show the alternative freq, which uses the tag frequency directly as the relevance level to compute NDCG scores, *i.e.* $s(i) = f_{d'}^{t}$.

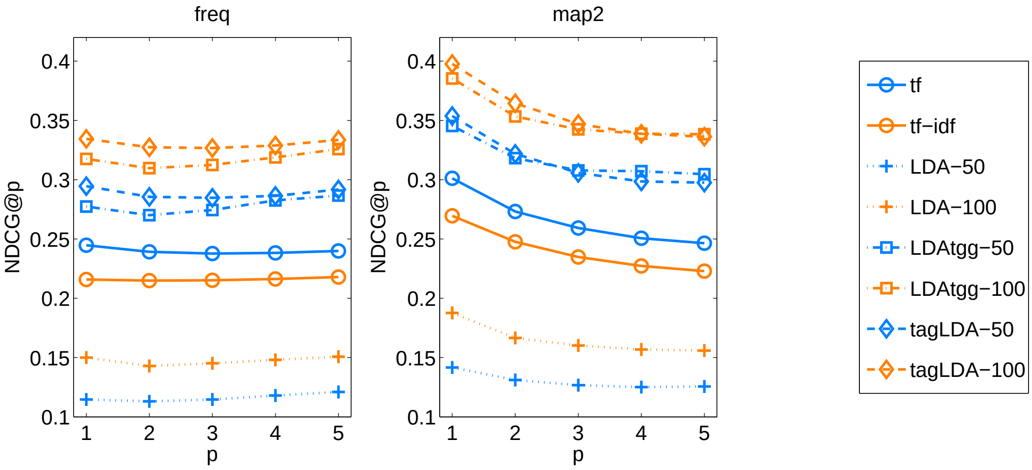


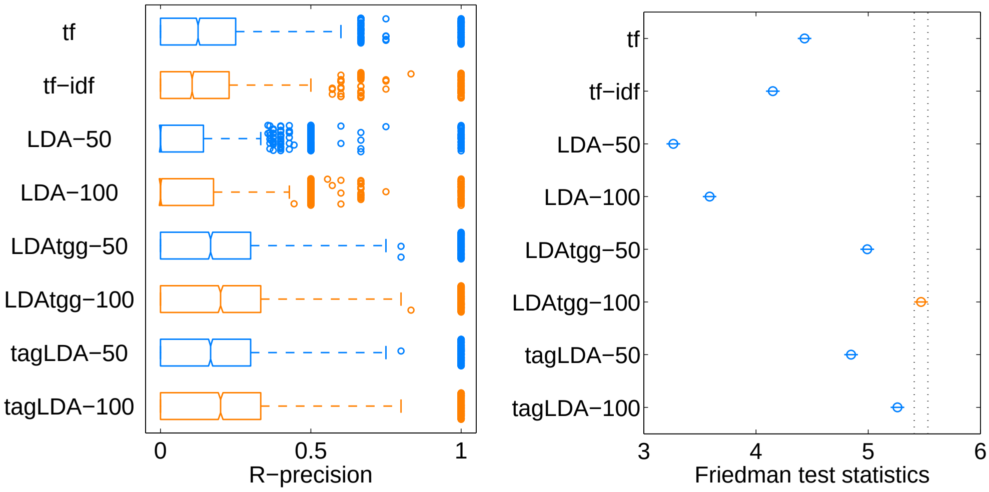
Figure 3.6: Tag Prediction Accuracy in NDCG

When using freq for scoring the ground truth tags, the NDCG@p values remain at almost the same level for $p \in [1, 5]$. Similar observations are also noted in the work by Au Yeung*et al.* [5], in which Au Yeung *et al.* attributed this effect to the skewness of the frequency distributions of the ground truth tags in documents. In other words, the frequencies of the top few tags are much higher than those less frequent ones. As a result, the normalization terms in NDCG@p becomes extremely large, and when the DCG@p increases, the amount of cumulative gain is largely discounted by the normalization terms. Exceptions are shown in LDAtgg-50 and LDAtgg-100 methods, in which the NDCG@p values begin to increase at $p \geq 3$. It suggests that, LDAtgg ranks more tags with higher relevance levels (cum frequency in this case) at higher p.

When using map2 for scoring the ground truth tags, the NDCG@p values are high at small p, and decrease as p increases. Moreover, LDAtgg-50 outperforms tagLDA-50 at $p \geq 4$, and LDAtgg-100 outperforms tagLDA-100 at $p = 5$. Figure 3.6 suggests that, even when the relevance levels of the ground truth tags are taken into account, the performance comparisons between methods do not differ from that in Figure 3.5.

R-precision and Significance Test

Figure 3.7.a shows the r-precision for all methods, measured for all test documents. We observe that all methods achieve r-precision=1 for some documents. In other words, the respective top n'_d predictions by the methods match all ground truth tags for the documents. We also note that, $n_{d'}$ are mostly small for these documents, *i.e.* in the range $[1, 4]$.



3.7.a: Measurements   3.7.b: Significance Test

Figure 3.7: Tag Prediction Accuracy in R-precision

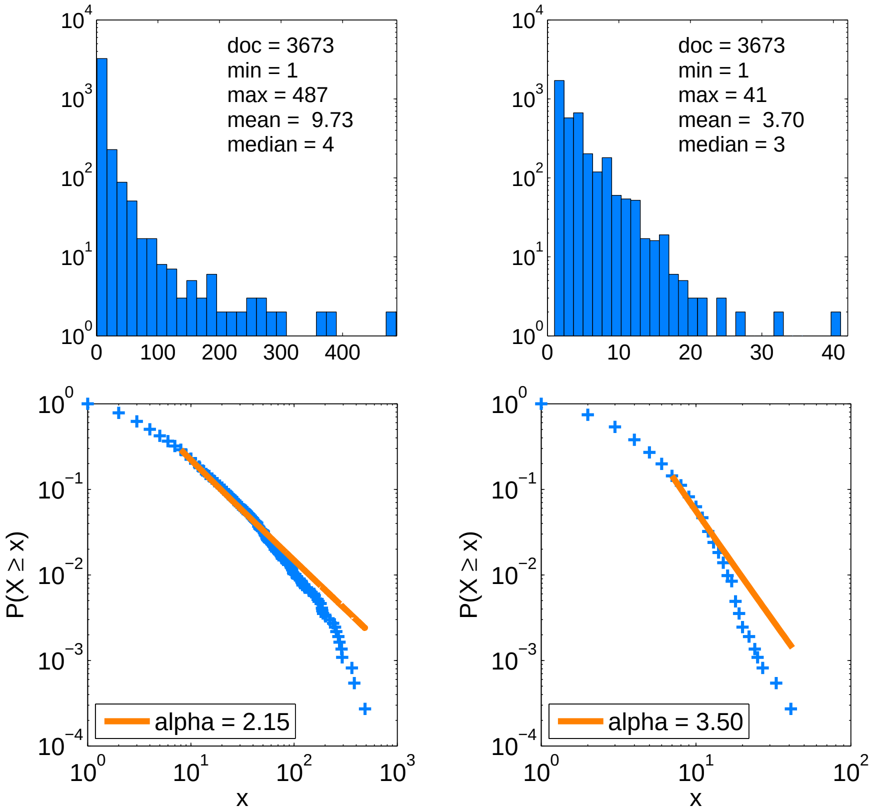
In Figure 3.7.a, although all methods are able to achieve r-precision=1 for some test documents, the median and inter quartile ranges still tell differences between the overall performances of the methods. On the whole, topic-based methods give higher median r-precision measurements than content-based methods. To test which method performs best and visualize its significance, we conduct Friedman multiple comparison test, as shown in the Figure 3.7.b. The use of Friedman multiple comparison test is suggested by

Hull [53]. Since the measurements on r-precision do not conform to the normality assumption, as we have verified in quartile-quartile plots (also known as QQ-plot), Friedman's test is more appropriate than multiway ANOVA test [53]. The Friedman multiple comparison test shows that LDAtgg-100 performs the best in r-precision, and all other methods fall to the left of the 95% confidence interval of the best performer. In other words, LDAtgg-100 is significantly better than all other methods, including its close competitor tagLDA-100.

3.5.2 Prediction Accuracy on Obvious Tags

In this section, we examine the methods' abilities to predict *obvious* tags. As noted by Farooq *et al.* [24], *obvious* tags are those appearing in the content of the target resource. These tags are obvious since they can be picked up from the document content for annotating the resources, as opposed to tag terms that are made up by the users [24, 46]. Note that methods such as tf and tf-idf are only able to predict tags that are obvious. This may be a disadvantage for tf and tf-idf, as the pool of candidate tags for these methods are smaller than those for the other methods. Hence, one may ask whether the good performance of the topic-based methods are mainly gained from predicting the *non-obvious* tags. Therefore, we conduct evaluation to compare the methods by the prediction accuracy on obvious tags.

We perform post hoc filter on both the ground truth tags and the tag predictions by the methods. For ground truth, we filter out the non-obvious tags. For topic-based methods, we filter out the predictions that do not appear in the content of the particular target document. As a result, the filtered predictions by topic-based methods include only the tags that appear in the tag vocabulary and appear as content words of the target document. For content-based methods, we filter out the predictions that do not appear in the tag vocabulary. As a result, the filtered predictions by content-based methods include the same set of tags as the predictions by topic-based methods.



3.8.a: Tag Tokens in Documents

3.8.b: Distinct Tags in Documents

Figure 3.8: Distributions for Obvious Tags in Documents

The post hoc filtering discarded from the evaluation a number of test documents that are only assigned non-obvious tags. For the remaining test documents, we plot the statistics of their ground truth tags in Figure 3.8. When comparing the statistics shown in Figure 3.8 with those in Figure 3.3, there are not only fewer documents, but also fewer tags for these documents. Now, a document has at most 41 distinct obvious tags, instead of up to 197 distinct assigned tags in Figure 3.3.b. This shows that a large proportion of tags are non-obvious keywords that do not appear in the documents. Moreover, the cumulative distribution for distinct tags in documents does not show as good power law fit as in the unfiltered counterpart in Figure 3.8.

We show in Figure 3.9 the the precision@p, recall@p and f1@p measurements of the various methods, computed based on the filtered ground truth and predictions. As Figure 3.9 shows, all methods show improved precision@p, recall@p and f1@p measurements than those in Figure 3.5. The performances

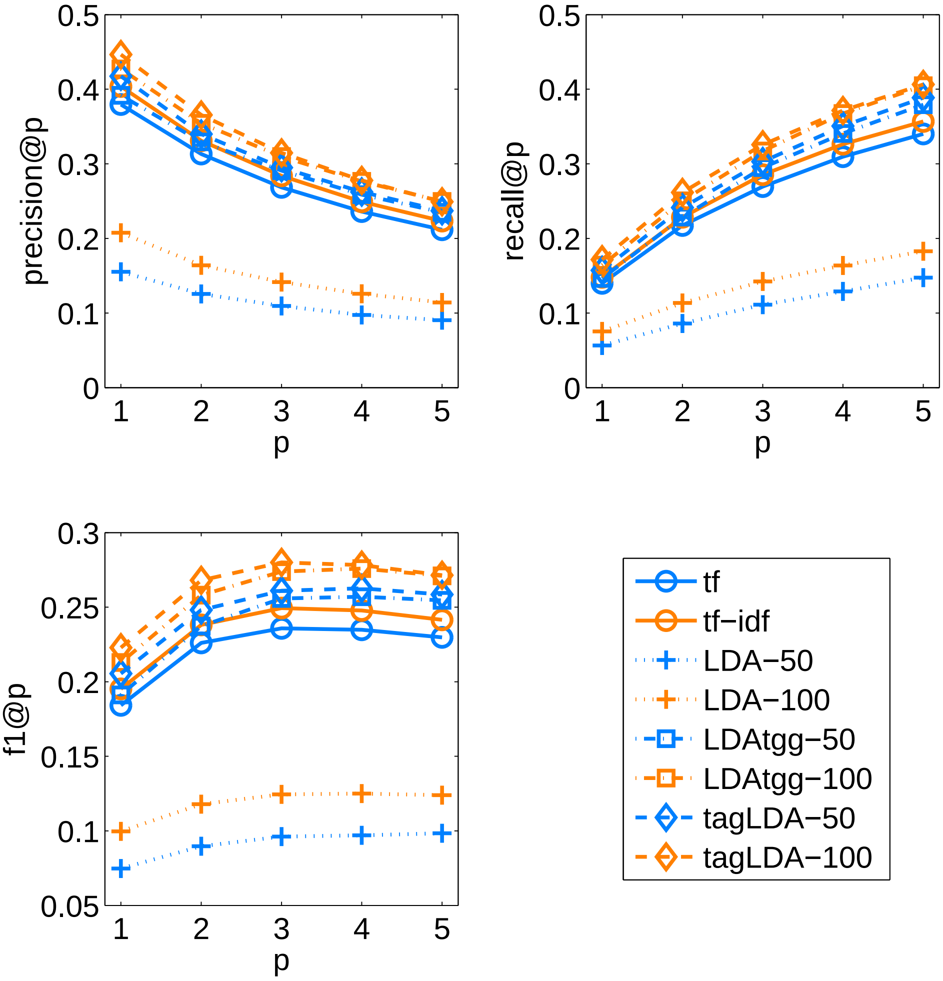


Figure 3.9: Tag Prediction Accuracy for Obvious Tags

of tf and tf-idf are still worse than those of the topic-based methods, but by a smaller margin. This is despite that we expect methods such as tf and tf-idf to show strong performance in predicting obvious tags. This observation suggests that topic-based methods is not merely good at predicting the non-obvious tags, but also good at ranking the obvious tags.

We also notice that tf-idf now outperforms tf in Figure 3.9, as opposed to the results shown in in Figure 3.5. This is because the rare content words, which were ranked higher by tf-idf, are now filtered out, since they do not appear in the tag vocabulary. After filtering out those rare content words as candidate tags, tf-idf gives better ranking than tf.

3.5.3 Prediction Accuracy vs. Characteristics of Documents

In this section, we examine the tag prediction accuracy of the methods with respect to characteristics of the test documents. In particular, we identify two characteristics for documents, namely *tag adequacy* and *tag exclusiveness*.

- We measure *adequacy* for each document as the total number of distinct tags in the ground truth for the document. Formally, $adq\left(d'\right) = n_{d'} = \sum_t I\left(d', t\right)$, where $I\left(d', t\right)$ is an indicator function that returns 1 if t appears as a ground truth tag for the test document d' and returns 0 otherwise.

- We measure *exclusiveness* for each document as the average exclusiveness of the distinct tags in the ground truth for the document. We define *exclusiveness* for each tag as its inverse document frequency in the entire dataset. Formally, $exc\left(d'\right) = \frac{1}{n_{d'}} \sum_t exc\left(t\right) \times I\left(d', t\right)$ and $exc\left(t\right) = \log \frac{D}{\sum_d I(t,d)}$.

Intuitively, we expect the methods to perform better for documents with more ground truth tags, and for documents with less exclusive tags. To conduct the comparison, we first partition the documents by the quartile ranges in the respective measurements, and then compute the average precision@5 for documents in each partition. Figure 3.10 shows the average precision@5 given by the methods when documents are partitioned by adequacy, and Figure 3.11 shows the same measurements for documents partitioned by exclusiveness.

In Figure 3.10, we observe that all methods give higher precision@5 for documents with higher adequacy, *i.e.* documents with more ground truth tags. This conforms to our expectation that tag predictions are easier for documents with more distinct tags, and harder for documents with fewer tags. The relative performance between methods in the different partitions does not differ from that in Figure 3.5.

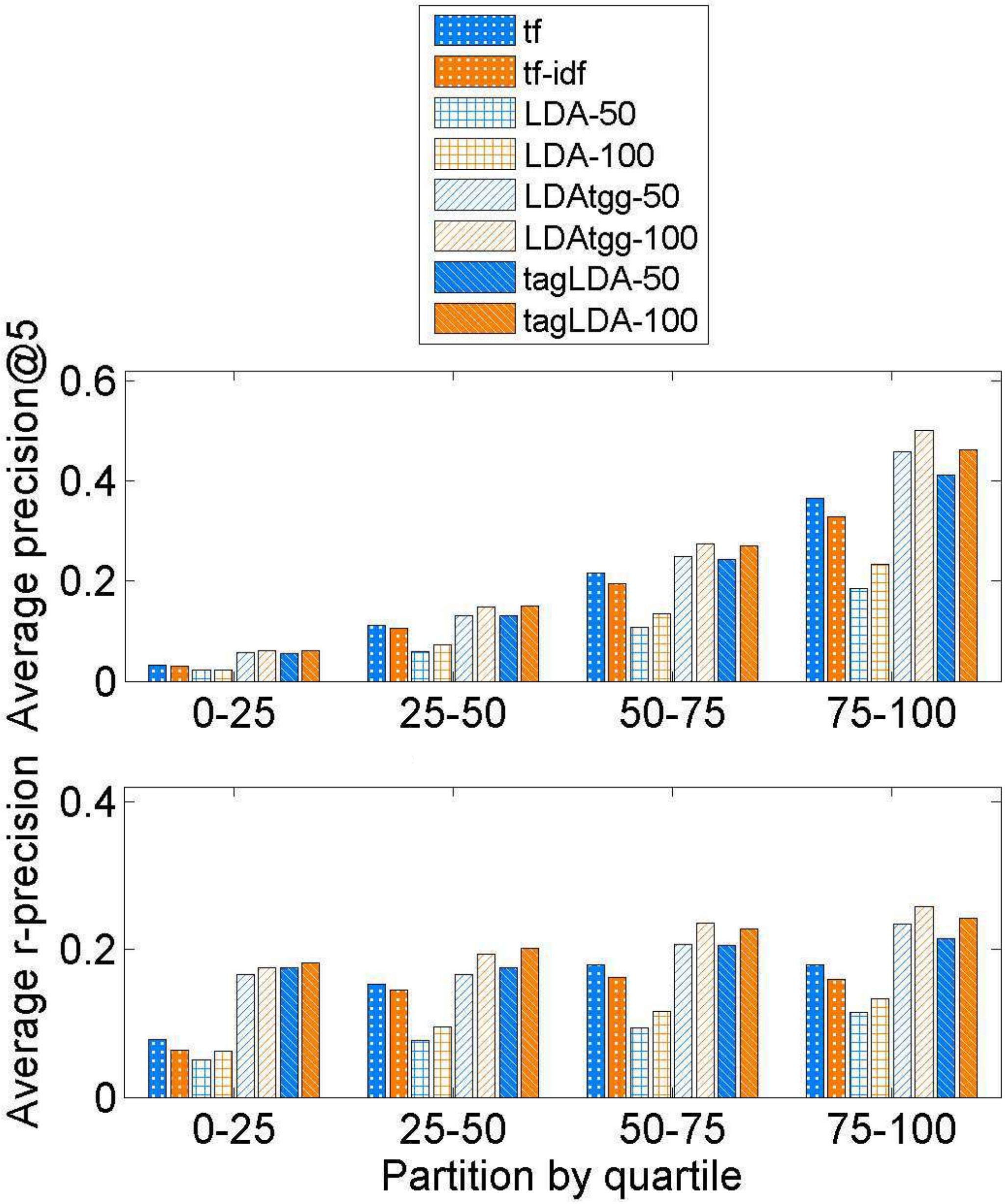


Figure 3.10: Tag Prediction Accuracy vs. Adequacy for Documents

In Figure 3.11, we observe that all methods give higher precision@5 for documents with lower exclusiveness of tags on average. The topic-based methods outperform the content-based methods by a large margin especially for the lowest exclusiveness partition. This observation suggests that topic-based methods are good at predicting tags that are less exclusive, *i.e.* tags assigned to more documents. For the partition of documents in the highest quartile, topic-based methods do not show much superior performance than content-based methods. For this partition, tf-idf also gives performance comparable to

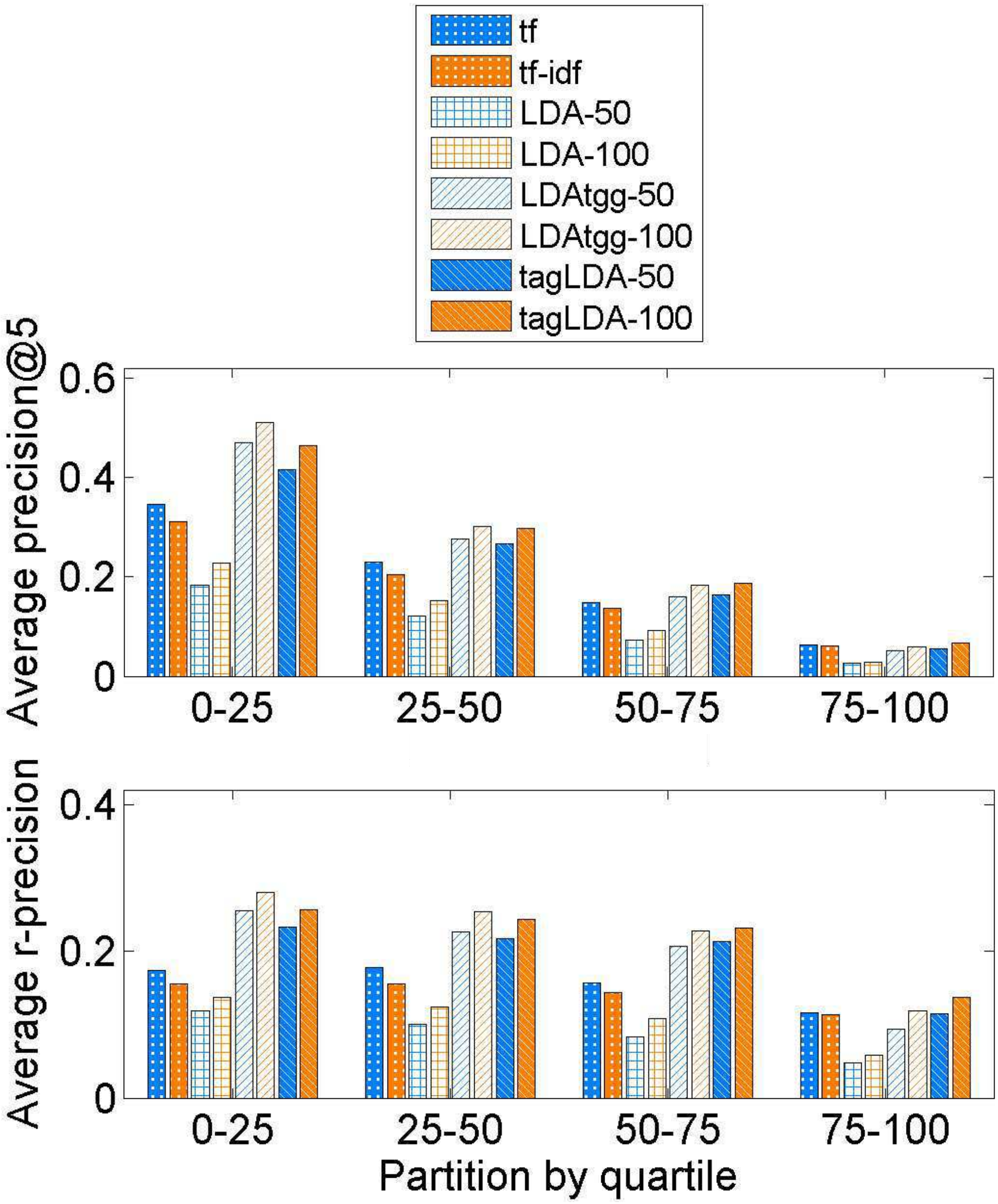


Figure 3.11: Tag Prediction Accuracy vs. Exclusiveness for Documents

that of tf, and tagLDA gives better performance than LDAtgg. This suggests that for documents with very exclusive tags, even though they are hard cases for all methods, methods such as tf-idf and tagLDA show some advantage for some cases.

Prediction Cases for Methods vs. Tag Exclusiveness

In this section, we sample some prediction cases for which one of the methods achieves high precision@5, *i.e.* precision@5= 1. Table 3.9 shows one prediction case for each method, denoted as the *best method*, and the top 5 predictions by all *other methods*. We highlight the correct predictions in boldface. For methods based on topic models such as LDA, LDAtgg and tagLDA, we select their 100-topic representatives to show the prediction cases. For each predicted tag t, we also show its corresponding $exc\,(t)$ values, which indicate the exclusiveness of the tag t in our dataset. Note that, higher $exc\,(t)$ indicates more exclusive tags, and lower $exc\,(t)$ indicates more common tags. If the predicted tag is a content word that does not appear in the tag vocabulary, particularly for methods such as tf, tf-idf and LDA-100, we mark the $exc\,(t)$ values for such predictions as '–'.

Table 3.9: Prediction Cases for Methods

Rank	Best Method		Other Methods							
	tf		tf-idf		LDA-100		LDAtgg-100		tagLDA-100	
	t	$exc\,(t)$	t	$exc\,(t)$	t	$exc\,(t)$	t	$exc\,(t)$	t	$exc\,(t)$
1	**activision**	7.717	**activision**	7.717	**games**	4.649	**games**	4.649	**games**	4.649
2	**blizzard**	7.717	**vivendi**	7.717	**game**	5.845	**gaming**	4.884	**gaming**	4.884
3	**games**	4.649	**blizzard**	7.717	video	3.598	**business**	3.240	technology	2.999
4	**vivendi**	7.717	**warcraft**	8.410	**business**	3.240	technology	2.999	**business**	3.240
5	**business**	3.240	**games**	4.649	world	3.623	online	4.106	google	5.114
URL	http://news.bbc.co.uk/2/hi/technology/7123582.stm									
Rank	Best Method		Other Methods							
	tf-idf		tf		LDA-100		LDAtgg-100		tagLDA-100	
	t	$exc\,(t)$	t	$exc\,(t)$	t	$exc\,(t)$	t	$exc\,(t)$	t	$exc\,(t)$
1	**pausch**	8.410	**pausch**	8.410	family	4.626	**news**	1.700	science	2.955
2	**lecture**	7.717	**lecture**	7.717	**people**	5.043	science	2.955	education	3.683
3	**mellon**	8.410	computer	4.604	years	–	**technology**	2.999	**cnn**	2.268
4	**carnegie**	8.410	fun	5.232	**life**	4.459	**cnn**	2.268	travel	3.078
5	**randy**	8.410	**carnegie**	8.410	time	6.331	history	3.174	family	4.626
URL	http://www.cnn.com/2008/SHOWBIZ/books/07/25/obit.pausch/index.html									
Rank	Best Method		Other Methods							
	LDA-100		tf		tf-idf		LDAtgg-100		tagLDA-100	
	t	$exc\,(t)$	t	$exc\,(t)$	t	$exc\,(t)$	t	$exc\,(t)$	t	$exc\,(t)$
1	**ice**	5.771	**ice**	5.771	**ice**	5.771	**science**	2.955	**climate**	4.626
2	**climate**	4.626	**arctic**	5.845	**arctic**	5.845	**globalwarming**	4.604	**environment**	3.520
3	**change**	5.078	scientists	7.717	hadow	–	**bbc**	2.035	**globalwarming**	4.604
4	**warming**	5.771	**expedition**	7.717	**expedition**	7.717	**climate**	4.626	**climatechange**	4.421
5	**global**	4.721	mr	8.410	sledge	–	**arctic**	5.845	**science**	2.955
URL	http://news.bbc.co.uk/2/hi/science/nature/7917266.stm									
Rank	Best Method		Other Methods							
	LDAtgg-100		tf		tf-idf		LDA-100		tagLDA-100	
	t	$exc\,(t)$	t	$exc\,(t)$	t	$exc\,(t)$	t	$exc\,(t)$	t	$exc\,(t)$
1	**environment**	3.520	**alaska**	6.619	fairbanks	–	ice	5.771	**environment**	3.520
2	**science**	2.955	university	7.024	**alaska**	6.619	change	5.078	**science**	2.955
3	**climate**	4.626	warming	5.771	summers	–	**climate**	4.626	**climate**	4.626
4	**climatechange**	4.421	cold	6.331	warming	5.771	water	4.827	**climatechange**	4.421
5	**globalwarming**	4.604	fairbanks	–	trees	7.717	years	–	health	2.426
URL	http://www.usatoday.com/weather/climate/2006-05-29-alaska-globalwarming_x.htm									
Rank	Best Method		Other Methods							
	tagLDA-100		tf		tf-idf		LDA-100		LDAtgg-100	
	t	$exc\,(t)$	t	$exc\,(t)$	t	$exc\,(t)$	t	$exc\,(t)$	t	$exc\,(t)$
1	**security**	3.867	malicious	7.717	malicious	7.717	**computer**	4.604	**security**	3.867
2	**internet**	3.527	programs	7.717	**symantec**	7.717	**online**	4.106	**internet**	3.527
3	**technology**	2.999	report	5.771	programs	7.717	**internet**	3.527	**technology**	2.999
4	**computer**	4.604	**symantec**	7.717	criminals	7.024	information	5.191	privacy	4.855
5	**computers**	4.721	code	7.024	detected	–	**security**	3.867	**computer**	4.604
URL	http://news.bbc.co.uk/2/hi/technology/7340315.stm									

From Table 3.9, we observe that the tags predicted by methods such as tf and tf-idf are more exclusive than the other methods on average. For example, for the BBC article *Video game giants in $18bn merger*, for which tf achieves precision@5= 1, both methods tf and tf-idf rank activation, blizzard, and vivend among the top 5 predictions, whose exclusiveness values are high. Method tf-idf also picks up warcraft, whose exclusiveness value is even higher. In contrast, methods LDA-100, LDAtgg-100 and tagLDA-100 all pick up games and business, though predicted correctly, whose exclusiveness values are lower. For this document, tf and tf-idf achieve better performance, partly due to that the true tags for this document contain content words that are not common to many other documents.

There are also cases where the true tags for the documents are commonly assigned to other documents. For those cases, we do observe better performance by topic-based methods. For example, for the USAToday article *Alaska the 'poster state' for climate concerns* and the BBC article *Computer viruses hit one million*, LDAtgg-100 and tagLDA-100 achieve good prediction performance respectively. Although the top 5 tags predicted by these methods have relatively low exclusiveness values, they find matches in the true tags for these documents. Among the top 5 tags predicted by tf and tf-idf for these documents, there are tags very specific to the particular document, but are not assigned as tags, such as fairbanks for the former document and malicious for the latter document.

The insights suggested by these document cases are twofold. Firstly, these ground truth tags are assigned by different users to the respective documents, and they demonstrate different levels of exclusiveness. Therefore, for documents annotated with more exclusive tags, tf-idf performs well, whereas for documents with less exclusive tags, topic-based methods perform well. Secondly, these ground truth tags are valid at the time of our data crawl, but will become incomplete ground truth as more users assign tags to these URLs after our crawl. Therefore, it is expected that as more tags are assigned, topical tags may appear for the former document and exclusive tags may appear for the others. This poses the possibilities and challenges in combining these two approaches for producing tag predictions that cover different levels of exclusiveness.

3.5.4 Summary of Error Types

In this section, we examine the errors made by the prediction methods. Since the evaluation for tag prediction compares exact matches of tag terms, even when a method predicts a tag of the same or similar meaning with one of the ground truth tags, it is still judged as an error. Upon examining on the false positive predictions of the various methods, we identified two types of errors due to this stringent judgement, namely *morphological variations* and *partial matches*.

Morphological variations A ground truth tag and a predicted tag have the following relationships: (i) singular *vs.* plural forms: *e.g.* `allergy` and `allergies`; and, (ii) noun *vs.* adjective of the same word root: *e.g.* `economy` and `economic`. To find these types of matches between tag terms, we experimented with a number of stemming and lemmatization methods, including Porter stemmer, Lancaster stemmer and the WordNet lemmatizer. We found using Lancaster stemmer gave reasonable matches between terms. Hence, the summary reported hereafter are based on the stem matching results produced by the Lancaster stemmer module of the NLTK Toolkit[10].

Partial matches A ground truth tag is a concatenation of multiple atomic terms, and one of the atomic terms matches a predicted tag, *e.g.* `behavior` *vs.* `behavior.intervention`, `behavioreconomics`, and `behaviormodification` etc. To split the concatenated tags into multiple atomic terms, a text segmentation algorithm can be applied. In our experiments, we applied the Viterbi segmentation algorithm [16, 26]. Running this algorithm usually requires a dictionary of the valid atomic terms, so as to compute the

[10] `www.nltk.org`

likelihood of the candidate Viterbi paths for splitting [26]. The accuracy of the segmentation results depends on the terms seen in the dictionary. For example, if `youtube` is not recognized as an atomic term, then the algorithm will suggest the most likely split to be `you` and `tube` since these two terms may have been seen in the dictionary. Therefore, in order to minimize over-splitting errors, we supplied a corpus consisting of the content of all documents in our dataset.

In Table 3.10 and Table 3.11, we summarize the number of occurrences of these types of errors made by the methods for documents in the five folds. Note that, the left half of each table shows the prediction instances (*i.e.* false positives), and the right half of the each table shows the true tag instances (*i.e.* false negatives). Each table also shows the total number of non-exact matching instances of the corresponding types found in each method.

On the whole, content-based methods make more of these types of errors than topic-based methods. This observation suggests that when users assign tags to annotate the news articles, they are likely to choose the word form according to their own preferences or create complex (non-atomic) tags by selecting important words from the news content. We also note that, even though the total counts for topic-based methods are smaller than those of content-based methods, the counts for individual tag instances are more skewed in topic-based methods than in content-based methods. Particularly, the tag `news` is found in 119 instances as partially matched false positives by `LDAtgg-100`, while there are 428 false positive instances by the method in total. In contrast, for method `tf`, among 797 false positive instances in total, the most frequent tag `north` is only found in 22 instances.

It is worth pointing out that the stemming procedure is not perfect for matching morphological variations. We do observe pairs of tags which actually have different meanings but were deemed as morphological variations, *e.g.* `book` *vs.* `booking`, and `new` *vs.* `news`. This is partly due to the Lancaster stemmer

Table 3.10: Statistics for Errors due to Morphological Variations

(a) tf

False Pos.	Count	False Neg.	Count
economic	15	jobs	13
job	13	economics	11
game	12	science	11
book	11	books	10
financial	11	finance	10
		Total Instances	688

(b) tf-idf

False Pos.	Count	False Neg.	Count
iraqi	11	iraq	14
afghan	11	afghanistan	13
israeli	10	israel	10
olympic	10	olympics	10
game	8	employment	9
		Total Instances	753

(c) LDA-50

False Pos.	Count	False Neg.	Count
economic	35	politics	32
financial	33	finance	31
political	32	jobs	28
job	28	cars	21
car	21	economics	21
		Total Instances	300

(d) LDA-100

False Pos.	Count	False Neg.	Count
economic	37	finance	30
financial	32	jobs	26
job	26	movies	24
movie	24	politics	24
political	24	cars	20
		Total Instances	399

(e) LDAtgg-50

False Pos.	Count	False Neg.	Count
politics	27	new	24
news	25	political	23
economy	23	economics	16
election	7	economic	7
finance	7	job	7
		Total Instances	166

(f) LDAtgg-100

False Pos.	Count	False Neg.	Count
news	26	new	25
politics	26	political	23
economy	22	economics	14
election	10	movie	10
movies	10	elections	9
		Total Instances	214

(g) tagLDA-50

False Pos.	Count	False Neg.	Count
economy	34	economics	25
politics	26	political	23
jobs	14	job	14
election	11	movie	11
finance	11	elections	10
		Total Instances	194

(h) tagLDA-100

False Pos.	Count	False Neg.	Count
economy	34	economics	24
politics	25	political	22
election	11	job	11
finance	11	movie	11
jobs	11	elections	10
		Total Instances	220

we adopted, which trims off more suffix than the Porter stemmer in general. For example, `economy` becomes `econom` and `economics` becomes `economi` when Porter stemmer is applied, but both become `econom` when Lancaster stemmer is applied. Lancaster stemmer allows us to find more meaningfully matched morph, *e.g.* `economy` *vs.* `economics` in the above case, but also produces mismatched pairs.

This set of results pose two related questions. The first is on the judgement of tag predictions. In most existing studies, tag predictions are judged based on exact matches with the ground truth. Such a stringent judgement would consider *morphological variations* and *partial matches* to be errors, even though the predicted tags are semantically relevant. There are studies that perform stemming or splitting on tags [6, 84], and evaluate the tag prediction accuracy based on the processed forms. However, no empirical comparison has been

Table 3.11: Statistics for Errors due to Partial Matches

(a) tf

False Pos.	Count	False Neg.	Count
north	22	climatechange	20
obama	17	northkorea	14
climate	14	barackobama	10
change	11	north_korea	9
job	9	globalwarming	7
		Total Instances	797

(b) tf-idf

False Pos.	Count	False Neg.	Count
north	16	northkorea	13
climate	14	climatechange	10
obama	12	north_korea	8
korea	11	barackobama	7
warming	7	hillaryclinton	5
		Total Instances	703

(c) LDA-50

False Pos.	Count	False Neg.	Count
change	40	climatechange	35
world	37	northkorea	17
obama	32	barackobama	12
north	26	globalwarming	11
global	18	climate_change	10
		Total Instances	450

(d) LDA-100

False Pos.	Count	False Neg.	Count
change	33	climatechange	33
obama	32	northkorea	16
world	27	globalwarming	14
north	25	videogames	13
global	20	barackobama	12
		Total Instances	533

(e) LDAtgg-50

False Pos.	Count	False Neg.	Count
news	117	bbcnews	26
obama	28	news_stories	16
health	21	barackobama	10
bbc	20	climatechange	8
climate	14	bbczimnews	7
		Total Instances	381

(f) LDAtgg-100

False Pos.	Count	False Neg.	Count
news	119	bbcnews	24
obama	29	news_stories	15
health	20	climatechange	13
climate	19	barackobama	10
election	18	election08	10
		Total Instances	428

(g) tagLDA-50

False Pos.	Count	False Neg.	Count
obama	31	climatechange	17
news	30	election08	14
climate	23	bbcnews	12
election	23	barackobama	11
health	23	news_stories	7
		Total Instances	370

(h) tagLDA-100

False Pos.	Count	False Neg.	Count
obama	29	climatechange	18
climate	24	barackobama	11
health	22	election08	11
election	21	bbcnews	8
korea	18	northkorea	7
		Total Instances	384

made between these design choices. The second question is on the performance comparison between the methods: if we take into account these types of errors and consider them correct predictions, will this invalidate the performance comparison observed in the previous sections? To answer this question, we further conduct a novel set of evaluation.

To evaluate the prediction accuracy by incorporating *morphological variations* and *partial matches*, we recompute precision@p, recall@p and f1@p measurements, but modify the judgement on tag predictions, defined as $I\left(d', t_i\right)$ in Eq. 3.9 and Eq. 3.10. Recall that t_i denotes the candidate tag ranked at position i by a method for a test document, and d' denotes the test document. Now, we set $I\left(d', t_i\right) = 1$ if and only if there exist a ground truth tag $\tilde{t}$ for d' such that:

1. t_i and $\tilde{t}$ are *exact match* or *morphological variations* or *partial matches*;

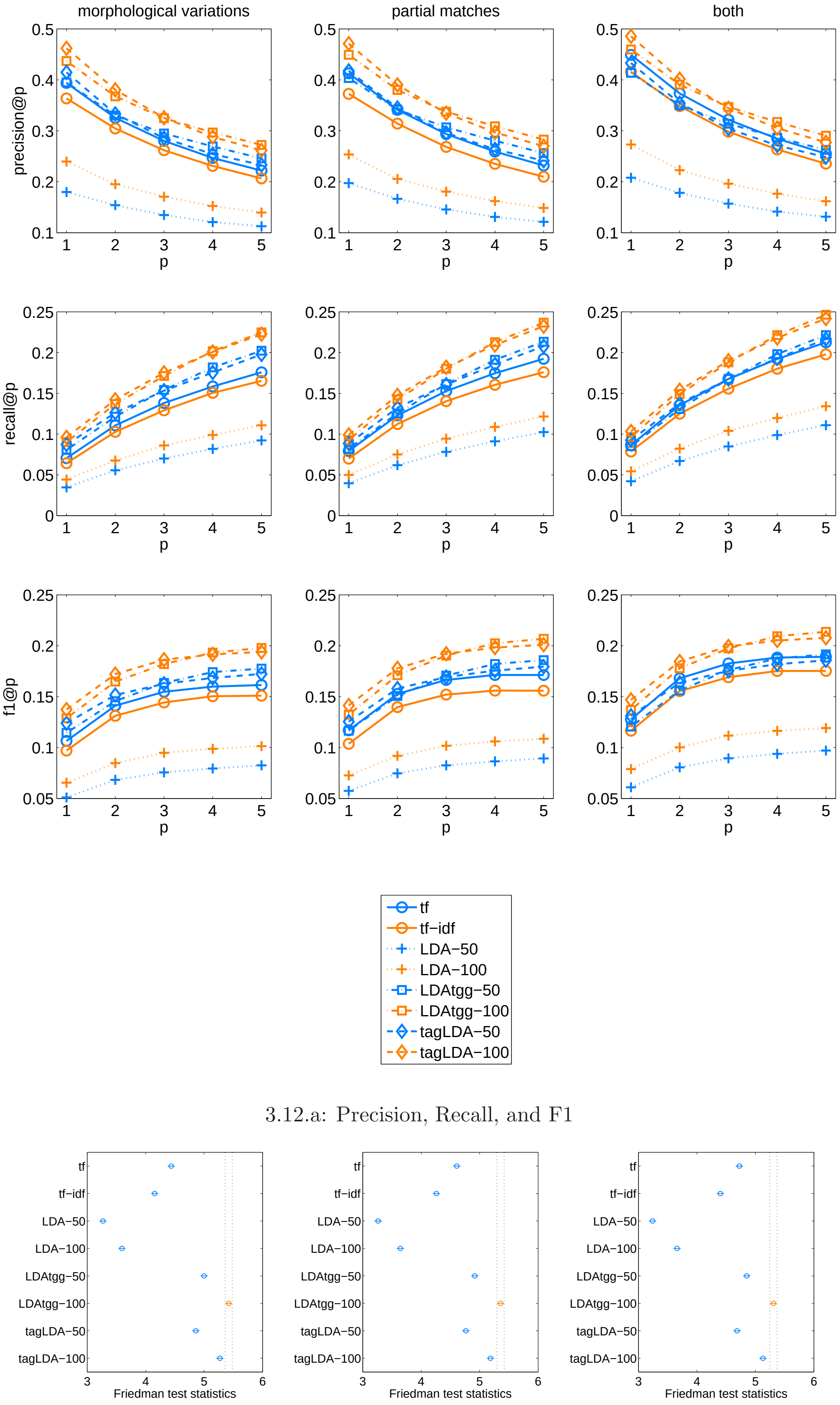
and

2. There does not exist t_j such that $j < i$ and t_j matches $\tilde{t}$ in any of the above forms.

The second condition is to avoid doubly counting a ground truth tag for which a match is found at a prior ranked position. For example, if a test document has ground truth tags such as `movie`, and the top predictions given by a method include `movies` and `movie`, we can only judge one of the morphological variations as relevant. In this case, `movies` is ranked prior to `movie`, when determining $I(d', t_i = \texttt{movie})$, since there exist $t_j = \texttt{movies}$ such that $I(d', t_j) = 1$, we will assign $I(d', t_i) = 0$. Note that based on the modified judgement, the precision@p and recall@p values computed according to Eq. 3.9 and Eq. 3.10 are still in the range $[0, 1]$. We plot the resulting accuracy measurements in Figure 3.12.a.

As expected, when *morphological variations* and *partial matches* are incorporated into the tag prediction judgement, the performances of content-based methods improved by a large margin as compared to that in Figure 3.5, especially for methods tf and tf-idf. When both types of non-exact matches are incorporated simultaneously, tf even outperforms LDAtgg-50 and tagLDA-50. Nonetheless, the two methods performed strongest previously, namely LDAtgg-100 and tagLDA-100, still outperform the others under this setting. Moreover, the same observations from Figure 3.5 still hold, *i.e.* tagLDA-100 performs better at lower p and LDAtgg performs superiorly at higher p.

Similar to Figure 3.7, we apply Friedman multiple comparison test on R-precision measurements, shown in Figure 3.12.b, 3.12.c and 3.12.d. We note that the Friedman test statistics for tf and tf-idf shifted rightward by a large margin as compared to Figure 3.7, but LDAtgg-100 remains significant over all other methods even when both types of non-exact matches are incorporated.



3.12.a: Precision, Recall, and F1

3.12.b: Morphological Variations

3.12.c: Parital Matches

3.12.d: Both

Figure 3.12: Tag Prediction Accuracy Incorporating Non-exact Matches

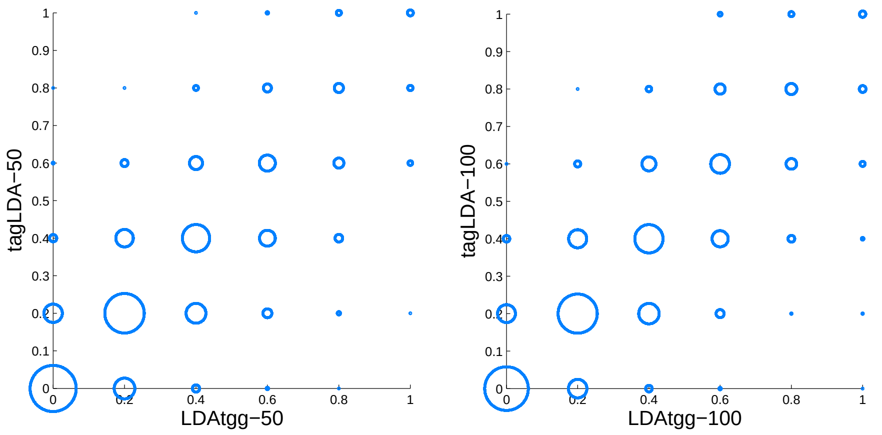
3.5.5 Comparison between Topic-based Methods

From the accuracy measurements shown in the previous sections, we observe that LDAtgg and tagLDA give competitive performances. In this section, we seek to answer the following questions:

1. Do the two models perform similarly for most of the documents? Or do they perform drastically different for most of the documents?
2. How are the performance differences related to the design of the models?

To answer the first question above, we plot the correlation on precision@5 between methods based on LDAtgg and tagLDA models, shown in Figure 3.13. The size of markers in the figure are proportional to the number of documents having the particular precision@5 values. We observe that the markers on the diagonal are of much larger size than those off the diagonal, and this holds for both $K = 50$ and $K = 100$. This suggests that LDAtgg and tagLDA produce the same precision@5 measurements for a large number of documents. Among the markers off the diagonal, those closer to the diagonal are of larger size than those more distant from the diagonal. This suggests that for these documents, the precision@5 measurements for LDAtgg do not differ drastically from that for tagLDA. There are only few documents for which methods based on these two models result in very large differences in prediction accuracy.

To answer the second question above, we identify the documents for which methods based on the two models perform contrastingly. We consider a method's performance poor if precision@5 ≤ 0.2, and strong if precision@5 ≥ 0.8. Tables 3.12 and 3.13 show two prediction cases, in which one document favors LDAtgg-100, *i.e.* demonstrating strong performance by LDAtgg-100 but poor performance by tagLDA-100, while the other document favors tagLDA-100, *i.e.* demonstrating strong performance by tagLDA-100 but poor performance by LDAtgg-100. In each table, we first list the URL of the document and the set of true tags of the corresponding document. We then report, for both LDAtgg-100



3.13.a: $K = 50$ 3.13.b: $K = 100$

Figure 3.13: Correlation between Topic-based Methods on Precision@5

Table 3.12: A Prediction Case Favoring LDAtgg-100

URL	`http://news.bbc.co.uk/2/hi/europe/7851292.stm`		
True Tags	article bbc cyber cyberattack cybersecurity cyberwar cyberwarfare defence dossier_otan eu europe european fis2304 frikifeeds internet it mct3328 nato news online p2p_warfare research security sota technology tietoyhteiskunta u.s. union www www_hack		
LDAtgg-100	**security bbc news technology internet**		
Chain 1	$p\left(k \mid d'\right)$	Words with High $p\left(w \mid k\right)$ for Topic	Tags with High $p\left(t \mid k\right)$ for Topic
	20.0%	security computer information software	security internet crime virus hacking
	9.7%	iraq troops afghanistan military war	iraq war military afghanistan terrorism
	5.5%	president meeting united policy deal	usa us world russia iran
Chain 2	$p\left(k \mid d'\right)$	Words with High $p\left(w \mid k\right)$ for Topic	Tags with High $p\left(t \mid k\right)$ for Topic
	21.3%	security information data computer	security privacy internet technology fbi
	12.2%	pakistan iraq afghanistan military troops	pakistan iraq afghanistan war military
	4.7%	europe european france french eu	europe russia france eu bbc
Chain 3	$p\left(k \mid d'\right)$	Words with High $p\left(w \mid k\right)$ for Topic	Tags with High $p\left(t \mid k\right)$ for Topic
	21.6%	security information computer data	security privacy surveillance nsa
	9.1%	afghanistan iran obama president policy	usa afghanistan iran politics obama
	4.5%	internet web online google bbc	internet google bbc news technology
tagLDA-100	**security** iraq war politics privacy		
Chain 1	$p\left(k \mid d'\right)$	Words with High $p\left(w \mid k\right)$ for Topic	Tags with High $p\left(t \mid k\right)$ for Topic
	18.8%	security computer software programs	security internet virus crime hacking
	11.0%	iraq afghanistan troops military war	iraq war military afghanistan usa
	5.3%	information data intelligence made privacy	privacy surveillance usatoday technology
Chain 2	$p\left(k \mid d'\right)$	Words with High $p\left(w \mid k\right)$ for Topic	Tags with High $p\left(t \mid k\right)$ for Topic
	16.5%	china chinese security computer	security china privacy politics internet
	11.7%	iraq afghanistan troops military war	iraq war military afghanistan iran
	3.4%	day time hours night sleep	sleep cnn funny stress misc
Chain 3	$p\left(k \mid d'\right)$	Words with High $p\left(w \mid k\right)$ for Topic	Tags with High $p\left(t \mid k\right)$ for Topic
	18.0%	security computer information data	security rss privacy politics internet
	9.0%	iraq afghanistan troops forces security	iraq afghanistan terrorism war military
	4.3%	europe european french eu france	europe france germany italy eu

and tagLDA-100, the top 5 predictions produced for the document, with correct predictions highlighted in boldface, the top topics (*i.e.* topic k's having highest $p\left(k|d'\right)$s) estimated from the document content , and the top words (*i.e.* word w's having highest $p\left(w|k\right)$) and the top tags (*i.e.* tag t's having highest $p\left(t|k\right)$) for the corresponding topics. We show the estimations for each document from the three Gibbs sampling chains, as noted in Section 3.4.2.

Table 3.13: A Prediction Case Favoring tagLDA-100

URL	http://www.cnn.com/2008/US/12/08/chicago.labor.protest/index.html		
True Tags	bankruptcy barack_obama chicago cnn currentevents economiccrisis economics economy illinois leftist manufacturing national politics recession stupidity text unions		
LDAtgg-100	**politics** news career jobs obama		
Chain 1	$p(k\|d')$	Words with High $p(w\|k)$ for Topic	Tags with High $p(t\|k)$ for Topic
	14.1%	job jobs workers work company	career jobs employment job work
	8.4%	clinton campaign democratic vote election	politics election elections campaign
	7.5%	financial money market mortgage banks	finance economics economy business
Chain 2	$p(k\|d')$	Words with High $p(w\|k)$ for Topic	Tags with High $p(t\|k)$ for Topic
	13.8%	cnn u.s. monday told tuesday	news commentary cnn satire mobius
	10.9%	bank financial money banks mortgage	finance economics business glossary crisis
	8.9%	job work workers jobs company	career jobs work employment job
Chain 3	$p(k\|d')$	Words with High $p(w\|k)$ for Topic	Tags with High $p(t\|k)$ for Topic
	13.5%	job work jobs workers company	career jobs work employment job
	12.5%	bill senate house republican secretary	politics government republicans opinion
	10.0%	money bank credit financial pay	money finance bank business banking
tagLDA-100	**economy politics recession** news **economics**		
Chain 1	$p(k\|d')$	Words with High $p(w\|k)$ for Topic	Tags with High $p(t\|k)$ for Topic
	10.4%	financial bank money banks mortgage	economics finance economy crisis business
	9.6%	people care american health americans	alex michelle ny 2008prezdebate3
	7.4%	economy economic jobs year industry	economy recession unemployment
Chain 2	$p(k\|d')$	Words with High $p(w\|k)$ for Topic	Tags with High $p(t\|k)$ for Topic
	12.0%	job workers jobs work company	career jobs work employment business
	9.1%	bank credit mortgage banks house	money finance bank banking banks
	7.1%	told cnn monday statement thursday	news satire mobius parody commentary
Chain 3	$p(k\|d')$	Words with High $p(w\|k)$ for Topic	Tags with High $p(t\|k)$ for Topic
	16.6%	jobs job workers people economy	economy recession unemployment greenjobs
	10.8%	financial bank banks money crisis	economy economics finance crisis business
	8.1%	illinois seat governor blagojevich senate	bank charges corruption clagg colson

For the document shown in Table 3.12, LDAtgg-100 achieves precision@5= 1.0, whereas tagLDA-100 achieves precision@5= 0.2. We observe that the tag security is correctly predicted by both methods. Upon a closer look at the topic mixtures estimated from the three Gibbs sampling chains, we see that security appears top, *i.e.* having highest $p(t|k)$, in one of the main topics, *i.e.* having highest $p(k|d')$, in all three chains. This holds for both LDAtgg-100 and tagLDA-100. The tag iraq is among the top 5 predictions by tagLDA-100, and it also appears top in one of the main topics in all three chains for this method. Unfortunately, iraq is not among the true tags of the document.

For this document, we note that the top topics, especially top 2 topics, are more consistent across the three chains in tagLDA-100 than in LDAtgg-100. Such consistency may sometimes benefit the predictions, *e.g.* for the case of security, and may also harm the predictions, *e.g.* for the case of iraq. Note that the final predictions produced by these topic-based methods are the combined predictions from the three chains. The predicted tags internet and technology are ranked above other candidate tags by LDAtgg-100, due to their high probabilities in the combined predictions from three chains.

For the document shown in Table 3.13, LDAtgg-100 achieves precision@5= 0.2, whereas tagLDA-100 achieves precision@5= 0.8. We observe that the tag politics is correctly predicted by both methods. We also note that the top topics estimated for the document are not as consistent across the three chains as for the document shown in Table 3.13. This holds for both methods LDAtgg-100 and tagLDA-100. For this document, tagLDA-100 predicts more relevant tags, such as economy and economics. Upon a closer look at the topic mixtures, these relevant tags come from the relevant topics, such as topic 57 in chain 1 and topic 76 in chain 2. tagLDA-100 also successfully identifies the topic on recession and unemployment, such as topic 25 in chain 1 and 43 in chain 2. Unfortunately, LDAtgg-100 fails to assign higher probabilities to these topics. It identifies the topic on jobs and career for this document, such as topic 19 in chain 1, topic 76 in chain2 and topic 95 in chain 3. Although the document content is relevant to these topic words, the top tags for these topics are less relevant. In this case, the tag unemployment should be more suitable than the tag employment.

We also note that for both documents, the tag news is among the top predictions by LDAtgg-100. As have shown in Table 3.6, news is one of the most frequent tags in our dataset. This tag also appears in the top predictions for many other documents. This observation suggests that, LDAtgg model is likely to predict tags seen more often in the training.

3.5.6 Run Time Measurements

We implemented our Gibbs sampler in MatLab, and executed our experiments on a Windows work station with Intel Core Duo CPU 2.33 GHz and 3.25 GB of RAM. Table 3.14 shows the run time measurements for training, re-sampling and prediction for each of the 5 folds, where measurements are averaged over the three independent chains. *h* denotes *hours* and *s* denotes *seconds*.

Table 3.14: Run Time of Topic-based Methods

K=50	Train		Re-sample		Prediction	
	LDAtgg	tagLDA	LDAtgg	tagLDA	LDAtgg	tagLDA
Fold 1	5.7482 h	3.7550 h	60.5303 s	60.4389 s	0.1680 s	0.1638 s
Fold 2	5.8221 h	3.8000 h	56.9431 s	57.2860 s	0.1776 s	0.1687 s
Fold 3	5.8984 h	3.8138 h	55.5393 s	55.8087 s	0.1716 s	0.1693 s
Fold 4	5.8201 h	3.7674 h	58.5637 s	58.3934 s	0.1772 s	0.1694 s
Fold 5	5.8337 h	3.7851 h	57.6721 s	57.3371 s	0.1731 s	0.1654 s

K=100	Train		Re-sample		Prediction	
	LDAtgg	tagLDA	LDAtgg	tagLDA	LDAtgg	tagLDA
Fold 1	12.3926 h	7.9981 h	122.3870 s	123.2093 s	0.2840 s	0.2526 s
Fold 2	12.5086 h	8.1168 h	115.8642 s	116.4983 s	0.2666 s	0.2581 s
Fold 3	12.6512 h	8.1947 h	113.3178 s	113.4284 s	0.2729 s	0.2611 s
Fold 4	12.4778 h	8.0700 h	119.8158 s	119.0074 s	0.2668 s	0.2588 s
Fold 5	12.5871 h	8.1213 h	116.5329 s	116.6661 s	0.2715 s	0.2534 s

These measurements show that the run time for Gibbs sampler is linear with respect to the number of latent topics, as the time taken for $K = 100$ almost doubles the time taken for $K = 50$. Although the training time is long, once the models are trained, predictions can be done efficiently. Note that the measurements for re-sampling and prediction are shown as the total time taken for all test documents in the respective fold. In other words, the time taken for each test document is much less, *e.g.* 0.0676 seconds on average to conduct re-sample and produce predictions when using LDAtgg-50. We expect the run time to increase as the number of latent topics increases.

LDAtgg and tagLDA take almost the same run time for re-sampling and prediction. Their run times differ at training. The time complexities of these two models using Gibbs sampler are the same. It is likely due to the power term in Eq. 3.2 that makes LDAtgg slower.

3.6 Summary

In this chapter, we discussed the task of tag prediction, which is an important task to address link sparseness between resources and tags in social tagging systems. For this task, we took a probabilistic topic-based approach to exploit the semantic relationships between resources, tags and content words. We proposed the LDAtgg model, which assumes *correspondence* on the distributions between the topics of tags and the topics of content words for the same resource. We developed a Gibbs sampling algorithm for learning model parameters, and conducted empirical evaluations using a novel dataset crawled from Delicious.

From our empirical evaluations, we found that tag prediction methods based on LDAtgg model gave superior performance than an extensive set of baseline methods. These baselines include content-based methods that selects content words from the target resource and a competitive topic-based method named tagLDA [18, 102]. tagLDA models the same set of variables as our LDAtgg model but assumes *conditional independence* between the topics of tags and the topics of content words. The tag prediction accuracy of LDAtgg is significantly higher than all other methods in terms of R-precision.

We make a number of observations. First, topic-based methods showed advantages over content-based methods, not only for all ground truth tags but also for *obvious* tags. Second, tag predictions are generally harder for documents with fewer ground truth tags and documents with more exclusive ground truth tags. Third, the tag predictions given by content-based methods and by topic-based methods are of different levels of specificity. In general, topic-based methods are likely to predict tags that are seen often in the (training) dataset, whereas content-based methods are likely to predict tags that are more exclusive to the target resource. Forth, we noted a number of prediction errors due to *morphological variations* and *partial matches*. However, even when these non-exact matches are incorporated into the evaluation of tag prediction accuracy, topic-based methods still showed superior performances than content-based methods. Lastly, when comparing the predicted tag by the two topic-based models, we found that the two models performed similarly for most of the documents but different for fewer documents. The tagLDA model has shown slight advantage in the average precision@5 for documents with very exclusive tags. Nonetheless, the Friedman multiple comparison test has shown that LDAtgg is significantly better in R-precision for all test documents.

The task of tag prediction is challenging on its own merit. One challenge faced by all tag prediction methods is the offline evaluation on the predicted tags. When more users bookmark the resources, tags which were not in the ground truth for a particular resource previously may become available. Therefore, for more thorough comparison between the tag prediction methods, online evaluation may be carried out. Nonetheless, there are ample room for improvements in the tag prediction methods studied in this chapter. For example, one may seek to combine topic-based methods with content-based methods to cover a broader range of specificity in the tag predictions. One may also seek to design criteria for selecting K for topic-based methods, or design a non-parametric topic-based model that does not require the number of topics to be pre-set. Studies continuing this research may explore these possibilities.

Chapter 4

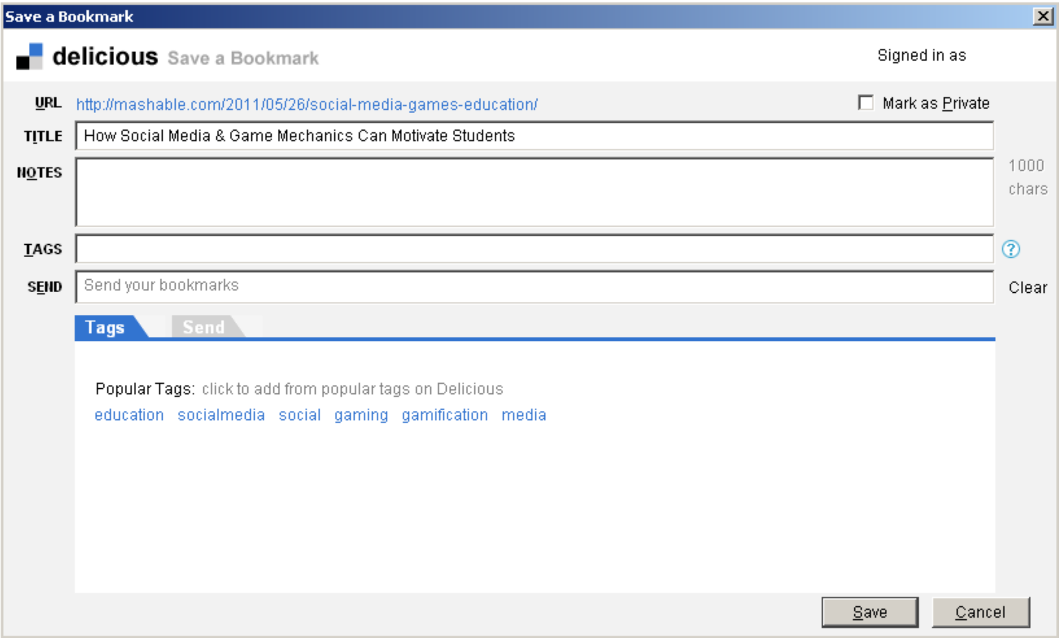
Personalized Tag Recommendation: A Probabilistic Framework

4.1 Introduction

Tag recommendation is an important task in social tagging systems. Tags are recommended at the time when a user initiates the annotation of a resource. Figure 4.1 shows the screenshot of Delicious[1] offering recommended tags to users when a resource is selected. Most social tagging systems recommend the most frequent (popular) tags that have earlier been assigned to the selected resource. While tags recommended as such can help consolidate the tag assignments across different users [34, 96, 99], the main purpose of tag recommendation is really to ease the annotation process for individual users. Therefore, it is important to recommend tags according to individual's tagging habits, because tagging activities are primarily for personal consumption [89, 78, 120].

Personal preference is prevalent in social tagging. For example, between synonyms, *e.g.* `Web` and `www`, users tend to be consistent in the choice of

[1] `www.delicious.com`



Save a Bookmark

delicious Save a Bookmark

Signed in as

URL http://mashable.com/2011/05/26/social-media-games-education/

□ Mark as Private

TITLE How Social Media & Game Mechanics Can Motivate Students

NOTES

1000 chars

TAGS

SEND Send your bookmarks

Popular Tags: click to add from popular tags on Delicious

education socialmedia social gaming gamification media

Figure 4.1: An Example of Tag Recommendation

tags, *e.g.* a user may always use `Web` or always use `www`, but not both at the same time. Intuitively, consistent tag assignments within personal collection of resources are better for organizing and retrieving the resources. Therefore, if a user prefers to use `Web` instead of `www` in annotating resources, a good tag recommendation algorithm should recommend `Web` when `www` is relevant in the context, so that resources related to `Web` and `www` are grouped under the same tag for this user. Since tag-based information organization and consumption is highly personal, personalized tag recommendations can better help the users organize the resources and increase the utility of the tagging data.

In our discussions, we refer to the selected resource for annotation as the *query resource*, and refer to the user to whom tag recommendations are provided as the *query user*. In social tagging systems such as Delicious, personalization in tag recommendation is done by simply matching the popular tags of the query resource with the existing vocabulary of the query user. Such recommendations based on tag popularity are not suitable for query users who do not follow the general crowd. Let us consider the following three scenarios:

(i) When the tag that the query user intends to use has only been previously used by very few other users to annotate the query resource.

(ii) When the tag that the query user intends to use has not been previously used by any other user on this query resource, but is in the query user's tag vocabulary.

(iii) When the tag that the query user intends to use has not been assigned to this query resource, nor has it been used by the query user herself, but it has been used by other users for annotating other resource(s).

Recommendations based on tag popularity fails to address scenario (i), because the intended tag is not a popular tag for the resource. However, re-scoring the existing tags of the resource may solve the problem. Collaborative filtering has been used to address scenario (i) [29, 77], but fails to handle scenario (ii), because the intended tag has not yet been used for the query resource. However, if one can translate from the existing tags of the query resource to the relevant tags in the personal vocabulary of the query user, it may solve the problem in this case. This method is known *personomy translation*. Personomy translation has been explored to address scenario (ii) [118, 120], but it fails to handle scenario (iii), because the intended tag has not been used by the query user in the past. For solving the problem, we seek to bridge the personomy translation of users.

In this chapter, we propose a probabilistic framework based on personomy translation that handles all three scenarios in a unified way. To reach out more candidate tags beyond the existing vocabularies of the query resource and the query user, we propose to leverage the personomy translations from other users. The solution we propose is inspired by the multilingual composition of the user population in social tagging systems. In the case of BibSonomy[2], a notable number of tags in German are observed besides the majority English tags. We also find that for resources with German tags (*e.g.* `foto`), the English equiv-

[2] `www.bibsonomy.org`

alent tags (*e.g.* `photo`) have also been assigned. Hence, we expect to see the German-speaking users sharing common translation patterns, *i.e.* translating from an English tag in the vocabulary of the query resource to a German tag in the vocabulary of the query user. This allows us to borrow the personomy translation performed by similar users for recommending tags relevant to the query user in the context.

In this research, we seek to answer the following questions:

(i) How to model personomy translations and use it for personalized tag recommendation?

(ii) How to identify users similar in personomy translation patterns?

(iii) How to incorporate the borrowed translation patterns into the tag recommendation algorithms?

(iv) Will personalized tag recommendation benefit from the borrowed translation patterns?

Our research contributions in this study can be summarized as follows:

- We solve the task of personalized tag recommendation as a probabilistic ranking problem, and propose a probabilistic framework for *neighbor-based translation* methods, which performs personomy translation and leverages the translation patterns from like-minded users.

- We propose to use distributional divergence metrics to measure the similarity between users in the context of personomy translation.

- We conduct experiments on a benchmark dataset collected from BibSonomy, and compare our proposed framework with baseline methods based on collaborative filtering and personomy translation by the query user solely. Our experimental results show that our neighbor-based translation methods outperform these baseline methods significantly. Moreover, we show that the translations borrowed from neighbors indeed help ranking relevant tags higher than that based solely on the query user.

4.2 A Probabilistic Framework for Personalized Tag Recommendation

In this section, we first introduce the essential concepts in a social tagging system and the notations to be used in our discussions. Next, we describe the probabilistic formulation on solving the tag recommendation task, and sketch a probabilistic framework that incorporates *personomy translation* from like-minded users. Lastly, we propose to use distributional divergence for measuring the similarity (dissimilarity) between users in the context of personomy translation, and discuss two variations in particular, namely JS-diversion and L1-norm.

4.2.1 Problem Definition

We represent a social tagging system as a *folksonomy* [34, 113], denoted by $\mathbb{F}$. A folksonomy consists of three types of objects, namely *resources*, *users* and *tags*, as well as the ternary relationships formed between these objects. Each relationship specifies a tag used by a user when she bookmarks a resource. Formally, we use R to denote a resource and $\mathbb{R}$ to denote a set of resources. We use U to denote a user and $\mathbb{U}$ to denote a set of users. We use T to denote a tag and $\mathbb{T}$ to denote a set of tags. A triplet $A = \langle R, U, T \rangle$ denote the relationship between R, U and T, and $\mathbb{A}$ denote the set of ternary relationships that exist in a folksonomy $\mathbb{F}$. We therefore have,

$$\mathbb{F} = \langle \mathbb{R}, \mathbb{U}, \mathbb{T}, \mathbb{A} \rangle \qquad \text{(Eq. 4.1)}$$

$$\mathbb{A} \subseteq \mathbb{R} \times \mathbb{U} \times \mathbb{T} \qquad \text{(Eq. 4.2)}$$

Table 4.1: Notations for Describing a Folksonomy

Symbol	Definition
R	a resource variable
U	a user variable
T	a tag variable
A	an assignment variable, denoting a ternary relationship $\langle R, U, T\rangle$
r	an instance of a resource
u	an instance of a user
t	an instance of a tag
a	an instance of an assignment relationship
$\mathbb{F}$	a folksonomy
$\mathbb{R}$	the set of all resource instances in the folksonomy
$\mathbb{U}$	the set of all user instances in the folksonomy
$\mathbb{T}$	the set of all tag instances in the folksonomy
$\mathbb{A}$	the set of all assignment instances in the folksonomy
$\mathbf{r}_u$	the set of resources in the subspace projection for u
$\mathbf{t}_u$	the set of tags in the subspace projection for u
$\mathbf{t}_r$	the set of tags in the subspace projection for r
$\mathbf{a}_u$	the set of assignments in the subspace projection for u

Table 4.1 lists these symbols and their definitions. For clarity and consistency, we use an uppercase letter to denote a variable and a lowercase letter to denote a particular value (instance) of a variable. We use a blackboard bold letter to denote the set of values for a variable.

Let us consider the example folksonomy shown as Figure 4.2. The set of resources are $\mathbb{R} = \{r1, r2, r3, r4\}$. The set of users are $\mathbb{U} = \{u1, u2, u3, u4\}$. The set of tags are $\mathbb{T} = \{t1, t2, t3, t4, t5, t6, t7, t8, t9\}$. The assignment relationships observed in this folksonomy include: $\langle r1, u1, t1\rangle$, $\langle r1, u1, t2\rangle$, $\langle r1, u1, t3\rangle$, $\langle r1, u2, t2\rangle$, $\langle r1, u2, t3\rangle$, $\langle r1, u2, t5\rangle$, $\langle r2, u2, t4\rangle$, $\langle r2, u2, t6\rangle$, $\langle r2, u2, t7\rangle$, $\langle r2, u4, t4\rangle$, $\langle r2, u4, t8\rangle$, $\langle r2, u4, t9\rangle$, $\langle r3, u2, t4\rangle$, $\langle r3, u2, t8\rangle$, $\langle r4, u3, t2\rangle$, and $\langle r4, u3, t5\rangle$.

One may project a folksonomy onto its subspaces. For example, given an instance of user, denoted by u, the subspace on u consists of the resources annotated by u (denoted by $\mathbf{r}_u$), the set of tags used by u (denoted by $\mathbf{t}_u$), as well as the set of assignment relationships specified by u (denoted by $\mathbf{a}_u$).

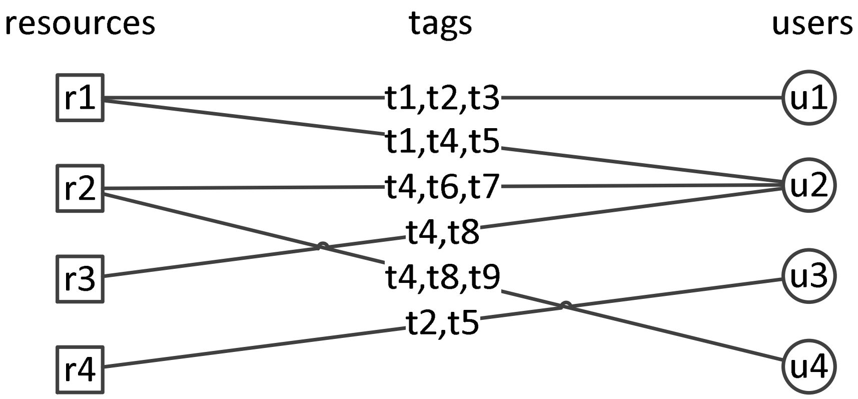


Figure 4.2: An Example of Folksonomy

Formally,

$$\mathbf{r}_u = \{r \in \mathbb{R} : \langle R, U, T \rangle \in \mathbb{A}, R = r, U = u\} \qquad \text{(Eq. 4.3)}$$

$$\mathbf{t}_u = \{t \in \mathbb{T} : \langle R, U, T \rangle \in \mathbb{A}, U = u, T = t\} \qquad \text{(Eq. 4.4)}$$

$$\mathbf{a}_u = \{\langle R, U, T \rangle \in \mathbb{A} : U = u\} \qquad \text{(Eq. 4.5)}$$

The subspace on u is also called the *personomy* of u [49, 118].

The tag recommendation task is to predict instances of the assignment relationship $\langle r, u, t \rangle$, when given the resource r and the user u. The input given to the recommender is a pair of query resource and query user, $\langle r, u \rangle_q$ (or equivalently $\langle r_q, u_q \rangle$). The expected output is the set of recommended tags that are relevant for describing the query resource by the query user, which we denote as $\{t\}_q$. Like an information retrieval task, the set of recommended tags are ranked by scores of relevance, $\delta(r_q, u_q, t)$.

4.2.2 A Probabilistic Framework

We solve the tag recommendation task as a probabilistic ranking problem. To compute the relevance score for a candidate tag, we estimate the likelihood of the tag given the pair of query resource and query user. Our main idea is that we can recommend a tag based on not only the query user's preference but also other like-minded users. We therefore formulate our probabilistic framework in Eq. 4.7.

$$\begin{aligned}\delta\left(r_q, u_q, t\right) &= p\left(t|r_q, u_q\right) && \text{(Eq. 4.6)}\\ &= \frac{\sum_u sim\left(u, u_q\right) \times p\left(t|r_q, u\right)}{\sum_u sim\left(u, u_q\right)} && \text{(Eq. 4.7)}\end{aligned}$$

The overall likelihood $p\left(t|r_q, u_q\right)$ of a candidate tag t is the weighted average of the likelihoods $p\left(t|r_q, u\right)$ from other similar users u, also known as *neighbors*, and the weight is the similarity between each neighbor u and the query user u_q, determined by $sim\left(u, u_q\right)$. The proposed framework is general and offers flexibility in three aspects. First, a neighbor (conceptually) can be the query user himself/herself or any other like-minded user in the social tagging space. The framework can assign the query user with the most weight. Second, many existing methods proposed in the literature [38, 57, 95, 118] can be adopted to estimate the likelihood $p\left(t|r_q, u\right)$, such as the graph-based method described in [38] and the tensor-based method described in [95]. Finally, the measure of similarity between users can also be varied, taking into account various aspects of the user profiles [77, 76].

In this study, for estimating the likelihood $p\left(t|r_q, u\right)$, we focus on the *personomy translation* methods proposed by Wetzker *et al.* [118, 120]. We first describe the methods in Section 4.2.3. We then introduce the distributional divergence metrics for measuring the (dis)similarity between users in the context of personomy translation in Section 4.2.4.

Table 4.2 summarizes the symbols used in our discussions.

4.2.3 Translating to Personal Preferences

The idea of personomy translation exploits the the relationships between a personal tag (*i.e.* $t \in \mathbf{t}_u$) of the user and the prior tags of the resources (*i.e.* $t_r \in \mathbf{t}_r$) [118, 120]. It estimates the likelihood of translating a resource tag to a personal tag of the query user, denoted by $p\left(t|u, t_r\right)$. A resource tag,

Table 4.2: Notations for Tag Recommendation Task

Symbol	Definition
r_q	the query resource (input) for tag recommendation
u_q	the query user (input) for tag recommendation
$\{t\}_q$	the set of recommended tags (output)
$\delta\left(r_q, u_q, t\right)$	the relevance score of t to the query resource and query user
$p\left(t \mid r, u\right)$	the likelihood of tag t given the resource r and the user u
$sim\left(u_1, u_2\right)$	the similarity between two users, *e.g.* u_1 and u_2
$p\left(t \mid u, t_r\right)$	the probability of translating t_r to t by the user u
$p\left(t_r \mid r\right)$	the probability of the tag t_r given the resource r
$sim_{JS}\left(X, Y\right)$	the similarity between two distributions X and Y, converted from JS-divergence
$sim_{L1}\left(X, Y\right)$	the similarity between two distributions X and Y, converted from L1-norm distance
$sim^{t_r}\left(u_1, u_2\right)$	the similarity between two users, u_1 and u_2, on their translation probability distribution for t_r
$p\left(t_r \mid u\right)$	the probability of the user u seeing the resource tag t_r

denoted by t_r, is a tag that has been assigned to the query resource prior to when the query user annotates the resource. A personomy tag, denoted by t, is a tag that has been used by the query user to annotate other resources. Wetzker *et al.* [118, 120] described two methods for estimating this likelihood based on tag-tag co-occurrences (*i.e.* (t, t_r)) perceived by the query user. We re-write both estimations in Eq. 4.9 and Eq. 4.10 respectively. Although [120] introduced a matrix-and-tensor based formulation, we provide a probabilistic view of the estimation in Eq. 4.10.

$$p\left(t \mid r_q, u\right) = \sum\nolimits_{t_r \in \mathbf{t}_r} p\left(t \mid u, t_r\right) \times p\left(t_r \mid r_q\right) \qquad \text{(Eq. 4.8)}$$

$$p\left(t \mid u, t_r\right) = \sum\nolimits_{r \in \mathbf{r}_u} p\left(t \mid r, u\right) \times p\left(r \mid t_r\right) \qquad \text{(Eq. 4.9)}$$

$$p\left(t \mid u, t_r\right) = \sum\nolimits_{r \in \mathbf{r}_u} p\left(t \mid r, u\right) \times p\left(t_r \mid r\right) \qquad \text{(Eq. 4.10)}$$

4.2.4 Measuring Preference Similarities

In the context of personomy translation, our hypothesis is that *users are similar to one another if they have similar translation patterns.* For instance, whenever annotating resources about `photo`, u_1 always assign `foto`, and whenever annotating resources about `Web`, u_1 always assign `netz`. So does u_2. In other words, u_1 and u_2 share common translation patterns, denoted by `photo`→`foto` and `Web`→`netz`. We therefore have reasons to believe that u_1 and u_2 are like-minded, and they may share even more common translation patterns in future tag assignments.

Formally, we say u_2 is similar to u_1, if their translation probabilities $p\left(t|u_1, t_r\right)$ and $p\left(t|u_2, t_r\right)$ are high and low together for different t and t_r pairs. Based on this intuition, we use *distributional divergence* to measure the similarity between users based on their translation probabilities.

Distributional divergence is the measure of distance between distributions. In this study, we describe and examine two distributional divergence metrics, namely *Jensen-Shannon divergence* and *L1-norm*, that are found useful in the literature [65]. Jensen-Shannon divergence (*JS-divergence* for short) is the symmetrized version of Kullback-Leibler divergence (*KL-divergence* for short). In information theory, KL-divergence is a measure for indicating the number of extra bits needed to represent the code samples in X using the code samples from Y, as compared to using the code samples from X itself. This interpretation fits our intuition of representing the translation probability from u_1 using the translation probabilities from u_2. However, KL-divergence is not a symmetric measure, which makes it not a true distance metric. Therefore, we use JS-divergence, which is symmetric:

$$D_{JS}\left(X, Y\right) = \frac{1}{2}\left[D_{KL}\left(X\|M\right) + D_{KL}\left(Y\|M\right)\right] \qquad \text{(Eq. 4.11)}$$

$$D_{KL}\left(X\|Y\right) = \textstyle\sum_i X\left(i\right)\log\frac{X(i)}{Y(i)} \qquad \text{(Eq. 4.12)}$$

$$M\left(i\right) = \frac{1}{2}\left(X\left(i\right) + Y\left(i\right)\right) \qquad \text{(Eq. 4.13)}$$

In Eq. 4.11, M is the average of the two distributions X and Y.

The L1-norm distance metric is written in Eq. 4.14. It is the sum of absolute distance between elements in the two distributions X and Y.

$$D_{L1}(X,Y) = \sum_i |X(i) - Y(i)| \qquad \text{(Eq. 4.14)}$$

To converte distance measure into similarity measure, we adopt the approach by Lee [64].

$$sim_{JS}(X,Y) = 10^{-\beta D_{JS}(X,Y)} \qquad \text{(Eq. 4.15)}$$

$$sim_{L1}(X,Y) = (2 - D_{L1}(X,Y))^{\beta} \qquad \text{(Eq. 4.16)}$$

The β in Eq. 4.15 and Eq. 4.16 are not equivalent. However, they have similar effect on the resulting measurements: higher β gives less importance to the distant neighbors. Following [64], we do not normalize the similarity scores across different metrics, even though they take different value ranges. For instance, $sim_{JS}(X,Y) \in [0,1]$ and $sim_{L1}(X,Y) \in \left[0, 2^{\beta}\right]$. The effect of value range will be cancelled out by the denominator in Eq. 4.7.

In personomy translation, each user has multiple sets of translation probabilities $p(T|u,t_r)$, one set for each t_r. Note that, $p(T|u,t_r)$ denotes a translation probability distribution. If two users have translation probabilities on a common t_r, we first measure the similarity between $p(T|u_1,t_r)$ and $p(T|u_2,t_r)$ using the metrics defined above. We use $sim^{t_r}(u_1,u_2)$ to denote this intermediate similarity measure. To derive the overall similarity between two users, we take the weighted average of $sim^{t_r}(u_1,u_2)$ on different t_r, and the weight is $p(t_r|u_1)$.

$$sim^{t_r}(u_1,u_2) = sim\left(p(T|u_1,t_r), p(T|u_2,t_r)\right) \qquad \text{(Eq. 4.17)}$$

$$sim(u_1,u_2) = \frac{\sum_{t_r} p(t_r|u_1) \times sim^{t_r}(u_1,u_2)}{\sum_{t_r} p(t_r|u_1)} \qquad \text{(Eq. 4.18)}$$

We interpret $p(t_r|u_1)$ as the likelihood of u_1 having seen t_r during tagging, *e.g.* t_r was assigned to the resource prior to u_1's annotation of the same resource.

This likelihood can be estimated from the tag collections of the resources that u_1 has annotated in the past.

$$p\left(t_r|u_1\right) = \frac{|\{\langle R,U,T\rangle \in \mathbb{A} : R = r \in \mathbf{r}_{u_1}, T = t_r\}|}{|\{\langle R,U,T\rangle \in \mathbb{A} : R = r \in \mathbf{r}_{u_1}\}|} \qquad \text{(Eq. 4.19)}$$

4.3 Dataset and Experimental Settings

We conduct experiments to demonstrate the effectiveness of the proposed probabilistic framework. We evaluate if the idea of borrowing personomy translation from similar users can expand the set of relevant tags beyond the existing tag vocabularies of the query resource and of the query user, hence getting better recommendation performance than the methods solely relying on the translations by the query user. We also compare with methods that are based on collaborative filtering [29, 77].

4.3.1 Data Preparation

Our datasets are collected from BibSonomy. Snapshots of BibSonomy have also been used as benchmark datasets in the ECML PKDD Discovery Challenge 2009 (DC'09 for short).

We create three datasets for our experiments, namely training, validation and test sets. We learn the translation probabilities and the similarities between users from the training set. We tune the parameters for optimal performance using the validation set. At last, we apply the optimal parameter settings when recommending tags for the query bookmarks in the test set.

We use the 2-core dataset provided in DC'09 as our training set. It is the snapshot of the BibSonomy before January 1, 2009. The dataset is *2-core* as every resource, user and tag appear in at least 2 bookmarks in this training set [49].

We take the *task2* dataset used for DC'09 contest as our validation set. All bookmarks in this validation set were made between January 1, 2009 and July 1, 2009, and only bookmarks for which the resources, the users and all the tags have appeared in the training set are included [49].

Our test set is taken from the most recent snapshot of BibSonomy, dated on January 1, 2010. This test set is collected by us, which was not used in DC'09. We follow the convention adopted in DC'09 for removing non-alphabetic and non-digit characters in the tags and normalizing them to their lowercase NFKC forms[3]. We extracted only query bookmarks that satisfy the following three requirements:

- The bookmarks are created between July 1, 2009 and January 1, 2010;
- The users of these bookmarks appear in the validation set;
- The resources and all tags in these bookmarks appear in the training set.

Table 4.3 shows the statistics of the three datasets.

Table 4.3: Statistics for BibSonomy Dataset

	Train	Validation	Test
Time frame	start date – 2008-DEC-31	2009-JAN-01 – 2009-JUN-30	2009-JUL-01 – 2009-DEC-31
Number of resources	22,389	667	258
Number of users	1,185	136	57
Number of distinct tags	13,276	862	525
Number of assignments	253,615	2,604	1,262
Average bookmarks per user	53.695	5.699	4.895
Average tags per bookmark	3.955	3.360	4.523
Average distinct tags per user	61.833	13.191	14.667

4.3.2 Experimental Setup

Evaluation Metrics

We adopt the *batch recommendation* setting, where tags are recommended *once* to the query user and query resource and these recommendations do not

[3] NFKC stands for Normalization Form Canonical Composition.

change regardless of any user input during the annotation process, as noted in Chapter 2. We adopt precision-recall curve and f1@5 as the main metrics for performance comparison and optimization. f1@5 is the harmonic mean of precision and recall at the 5-th position in the ranked list of recommended tags for a query post. f1@5 is also the evaluation metric used in DC'09 for the contest.

To define the evaluation metrics, we use t_i to denote the tag at position i in the ranked list of recommended tags, n_q to denote the total number of ground truth tags for the query bookmark, and p to denote the position in the list of recommended tags at which the evaluation takes place. Hence,

$$\text{precision@p} = \frac{\sum_{i=1}^{p} I_q(t_i)}{p} \qquad \text{(Eq. 4.20)}$$

$$\text{reall@p} = \frac{\sum_{i=1}^{p} I_q(t_i)}{n_q} \qquad \text{(Eq. 4.21)}$$

$$\text{f1@p} = \frac{2 \times \text{precision@p} \times \text{recall@p}}{\text{precision@p} + \text{recall@p}} \qquad \text{(Eq. 4.22)}$$

where the function $I_q(t_i)$ returns 1 if t_i matches one of the ground truth tags for the query bookmark and 0 otherwise.

We compute the metrics at $p \in [1, 5]$ for each query bookmark in the test set. To gain a user-centric view of tag recommendation performance, we compare the *macro-average* performance of methods. Macro-average is the average of the per-user average performances.

Methods Compared

We evaluate our proposed probabilistic framework by including three groups of methods.

trans-n1 and trans-n2: Both methods follow our proposed probabilistic framework in estimating the likelihood $p\,(t|r_q, u_q)$. We use letter n to indicate the inclusion of translations from neighbors. The two variations differ in the estimation of $p\,(t|u, t_r)$. trans-n1 follows Eq. 4.9, and trans-n2 follows Eq. 4.10.

We compute the similarities between users based on the estimated $p(t|u, t_r)$ for these two variations accordingly. When computing the similarity between users, there are two parameters to be determined: (i) β for converting the distributional divergence measure into similarity measure; (ii) k for selecting the number of nearest neighbors. For β, we explore in the range $\beta \in \{1, 2, 4, 8\}$ for JS-divergence and $\beta \in \{1, 2, 4, 8, 12, 16\}$ for L1-norm. For k, we explore in the range $k \in \{5, 10, 20, 50, 100, 200, 300, 400, 500\}$.

trans-u1 and trans-u2: These methods are special cases of the proposed framework. They remove other users when estimating $p(t|r_q, u)$. In other words, they rely on the translation probabilities estimated solely for the query user, but do not borrow translation from neighbors. We use letter u to indicate such distinction from the trans-n methods. For the estimation of $p(t|u, t_r)$, trans-u1 follows Eq. 4.9, and trans-u2 follows Eq. 4.10 respectively.

knn-ur and knn-nt: These methods are direct application of collaborative filtering to tag recommendation in folksonomies [29, 77]. They first select the k-nearest neighbors (knn for short) for the query user and recommend tags that have been assigned by the neighbors to the query resource. The overall relevance score of a candidate tag is the average similarity of the corresponding neighbors who have assigned the tag. The two variations differ in profiling the users for computing the similarity between users. In knn-ur, each user is represented as a vector of resources, and the vector weights are binary-valued to indicate whether the user has annotated each of the resources. Whereas in knn-ut, each user is represented as a vector of tags. The vector weights are the frequency of tags that have been used by the user[4]. The similarity between users is then computed as the cosine similarity in vector space. There is one parameter to be determined in this group of methods: k for selecting the number of nearest neighbors. We explore k in the same range as that for

[4] We have also tried using binary-valued weights in the user-tag representation. However, it shows similar performance with that using frequency-valued weights. Therefore, we do not include the binary-valued variation of this method.

trans-n methods, *i.e.* $k \in \{5, 10, 20, 50, 100, 200, 300, 400, 500\}$.

Finally, we also include the baseline method freq-r, as shown in Eq. 4.23. It recommends tags based on the frequency in which the tag has been assigned to the query resource. The underlying assumption is that, *the more often a tag has been assigned to the resource, the more likely it would be used again.*

$$p\left(t|r_q, u_q\right) = \frac{|\{\langle R, U, T\rangle \in \mathbb{A} : R = r_q, T = t\}|}{|\{\langle R, U, T\rangle \in \mathbb{A} : R = r_q\}|} \qquad \text{(Eq. 4.23)}$$

Although not performing personalization itself, freq-r has been reported to work well for tag recommendation tasks [29], especially when combined with methods that do perform personalization [118]. For exploring the performance space, we also combine freq-r with methods listed above. We adopt linear interpolation when calculating the interpolated likelihood of a candidate tag $p\left(t|r_q, u_q\right)$, shown in Eq. 4.24.

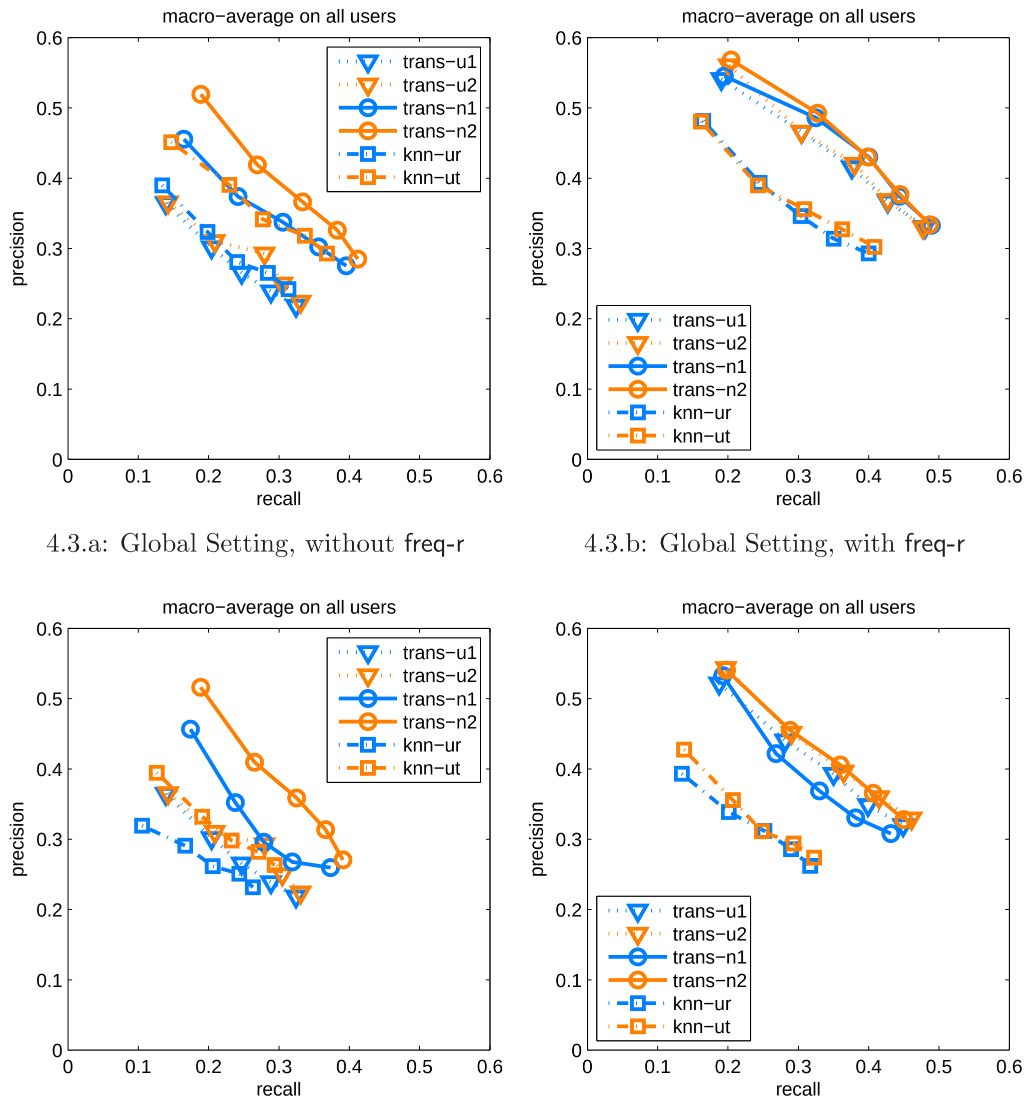
$$\begin{aligned} & p_{\mathsf{interpolated}}\left(t|r_q, u_q\right) \\ = \; & \omega \times p_{\mathsf{freq\text{-}r}}\left(t|r_q, u_q\right) + (1 - \omega) \times p\left(t|r_q, u_q\right) \end{aligned} \qquad \text{(Eq. 4.24)}$$

where ω is an additional parameter to be tuned in the interpolated estimations.

4.4 Experimental Results

4.4.1 Tag Recommendation Accuracy

We first examine the precision-recall curve (PR-curve for short) of the six recommendation methods listed in Section 4.3.2, with and without interpolated with freq-r. We then look at the macro-average f1@5 of the methods.



4.3.a: Global Setting, without freq-r

4.3.b: Global Setting, with freq-r

4.3.c: Individual Setting, without freq-r

4.3.d: Individual Setting, with freq-r

Figure 4.3: PR-Curve for Tag Recommendation on the Test Set

PR-curve on the Test Set

Figure 4.3 shows the performance on the test set in PR-curve, for which the corresponding optimal parameter settings are determined by the validation set. *Global setting* refers to applying the same set of parameters to all users, which have been tuned to optimize the macro-average f1@5 on the validation set. *Individual setting* refers to individualized parameters that optimize the average f1@5 for each user on the validation set. L1-norm metric is used for trans-n1 and trans-n2.

Without interpolated with freq-r, trans-n methods show clearly large advantage over trans-u methods. This holds for both global and individual settings. This consolidates our intuition that borrowing translations from similar users is able to help recommending more relevant tags to the query user for the query resource. On the whole, trans-n2 performs stronger than trans-n1. trans-n2 performs the best on the test set.

knn-ur always outperforms knn-ut. This observation is consistent with those made in [29, 77]. It suggests that users who are similar in their tag vocabularies are more likely to assign same tags(s) to the same resource, than those who are similar in their collections of annotated resources.

When interpolated with freq-r, all methods, except knn-ut, give largely improved performance over their non-interpolated counterparts. The performance by knn-ur is brought closer to that by knn-ut. However, the interpolated trans-u and trans-n outperform knn methods by an ample margin. This can be explained by the composition of candidate tags of knn methods. knn methods always recommend tags that have already been assigned to the query resource, in this case, by the k-nearest neighbors. In other words, the candidate tags of knn is a subset of that for freq-r. Hence, freq-r brings little additional benefit to knn-ut when the interpolation parameter ω is optimized. On the contrary, both trans-u and trans-n methods are able to bring non-existing tags to the query resource. These non-existing tags, some of which are indeed adopted by the query user to annotate the query resource, contributes to better performance for the translation based methods over freq-r and knn methods.

Although not performing well by themselves, trans-u1 and trans-u2 methods achieve large improvement when interpolated with freq-r. The candidate set of trans-u methods includes all tags that have been used by the query user in the past, be it relevant or less relevant to the current query resource. Applying trans-u methods alone may recommend highly personal tags that are less relevant to the current query resource. However, when interpolated with freq-r, tags that are relevant to the resource can be included as recommendations.

Therefore, we observe significant lift in the performance by trans-u1 and trans-u2 when interpolated with freq-r using optimized parameter settings.

To our surprise, individual setting does not outperform global setting on the test set. To find the explanation, we next examine the macro-average f1@5 achieved for the validation set as well as for the test set.

F1@5 on the Validation and Test Set

Table 4.4 and 4.5 show the macro-average f1@5 of the settings for the corresponding methods on the validation and test set respectively. The best performer within each column are highlighted in boldface. Again, L1-norm metric is used in trans-n1 and trans-n2 methods. Note that, the optimal parameter settings are obtained by optimizing the macro-average f1@5 on the validation set, and these settings are applied to the test set without re-optimization. In individual setting, they are optimized for each individual user. Whereas in global setting, they are optimized for all users.

Table 4.4: Macro-average f1@5 for the Validation Set

	Global setting		Individual setting	
	without freq-r	with freq-r	without freq-r	with freq-r
trans-u1	0.238	0.363	0.238	0.401
trans-u2	0.244	0.363	0.244	0.401
trans-n1	0.301	0.366	0.342	0.429
trans-n2	**0.310**	**0.367**	**0.359**	**0.430**
knn-ur	0.248	0.317	0.264	0.347
knn-ut	0.294	0.321	0.303	0.348

Table 4.5: Macro-average f1@5 for the Test Set

	Global setting		Individual setting	
	without freq-r	with freq-r	without freq-r	with freq-r
trans-u1	$^{\vee}$0.238	0.359	$^{\vee}$0.238	$^{\vee}$0.344
trans-u2	$^{\vee}$0.244	0.358	$^{\vee}$0.244	**0.354**
trans-n1	0.298	**0.363**	0.281	$^{\vee}$0.330
trans-n2	**0.310**	0.362	**0.293**	0.349
knn-ur	$^{\vee}$0.248	$^{\vee}$0.312	$^{\vee}$0.222	$^{\vee}$0.260
knn-ut	0.290	$^{\vee}$0.321	0.244	$^{\vee}$0.263

In Table 4.4, we observe that trans-n2 is consistently the best performer in all settings. Individual setting always achieves higher (or equal) average f1@5 than the corresponding global setting counterpart. This set of results matches our expectation, since not all users have equal number of like-minded neighbors or prefer equal number of resource tags, thus the parameter settings tailored to individual users should outperform the parameter settings equally applied to all users. This set of results also reaffirms the strong performance of trans-n2 method.

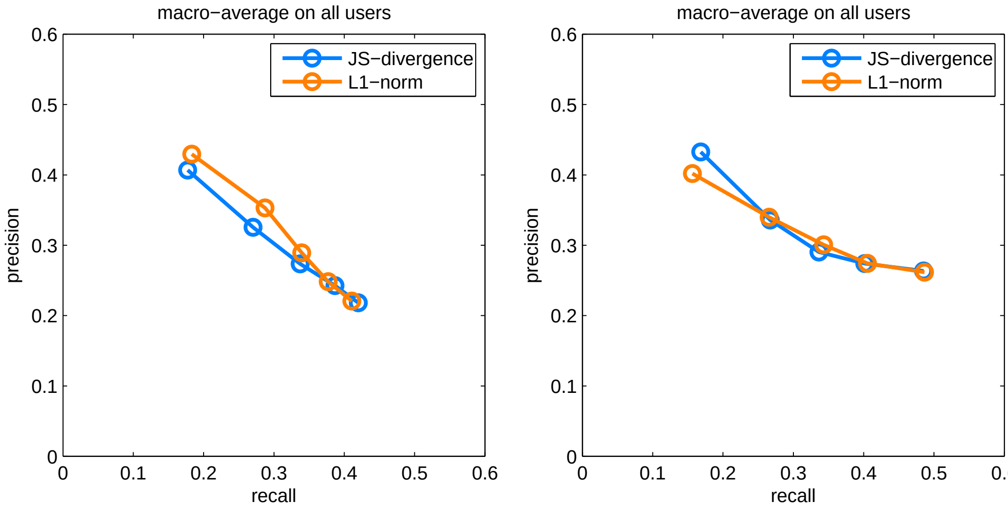
In Table 4.5, we observe that the best performer are not consistent across the different columns. Method trans-n2, however, is the best performer in both global and individual settings without interpolated with freq-r. When interpolated with freq-r, the interpolated trans-n1 is the best performer under global setting, and the interpolated trans-u2 is the best performer under individual setting.

To compare the best performer against the rest of the methods in each column, we conduct paired right-tail t-test with significance level of 0.05. We put a $\triangledown$ besides the macro-average f1@5 value of the *non-best-performing* method if the t-test indicates that the best performer outperforms the method significantly. We note that, when trans-n2 is the best performer, it always outperforms trans-u methods significantly. When trans-u2 is not the best performer, it shows very close performance to the best performer and its disadvantage is not significant.

By comparing Table 4.4 with Table 4.5, we find that even when the parameter settings are optimized on the validation set, these settings do not guarantee similar performance on the test set. This contrast is more obvious for individual settings. One reason for the downgraded performance on the test set is that users' tagging preferences change over time. As we noted in Table 4.3, when comparing the statistics on the validation set with those on the test set, users assign more tags per bookmark and use more distinct tags. It remains a research question on what other optimization criteria are suitable in the context, *e.g.* precision@1 and area under the croc-curve [98]? Future research on the same task may explore these questions.

4.4.2 Effect of the Divergence Metrics

We observe little performance difference due to the divergence metrics when the respective parameters are optimized. In Section 4.2.4, we have introduced two divergence metrics for measuring the divergence between users in the context of personomy translation, namely JS-divergence and L1-norm. Figure 4.4 shows the pr-curves by trans-n2 when using these two divergence metrics. Under both global and individual settings, the performance by the two metrics are close, though L1-norm shows slight overall advantage. Similar observation are made when trans-n1 is used. Therefore, we report the performance by trans-n1 and trans-n2 using L1-norm metric only in Figure 4.3 and Table **??**.



4.4.a: Global Setting, without freq-r   4.4.b: Individual Setting, without freq-r

Figure 4.4: Effect of Divergence Metrics

4.4.3 Parameter Tuning

Lastly, we look at the parameter settings tuned on the validation set. For global setting, we optimize the macro-average f1@5, and the chosen parameters by methods are shown in Table 4.6. For individual setting, we optimize the average f1@5 for each user, and plot the distribution of the chosen ω, β and k by users for the interpolated trans-n2 in Figure 4.5.

Table 4.6: Global Setting Tuned on the Validation Set

Method	Metric	Without freq-r		With freq-r		
		β	k	ω	β	k
trans-n1	l1	4	100	0.5	16	100
	js	2	200	0.5	8	300
trans-n2	l1	4	100	0.3	8	100
	js	2	200	0.1	4	400
knn-ur	–	–	400	0.9	–	200
knn-ut	–	–	400	0.9	–	200
trans-u1	–	–	–	0.4	–	–
trans-u2	–	–	–	0.4	–	–

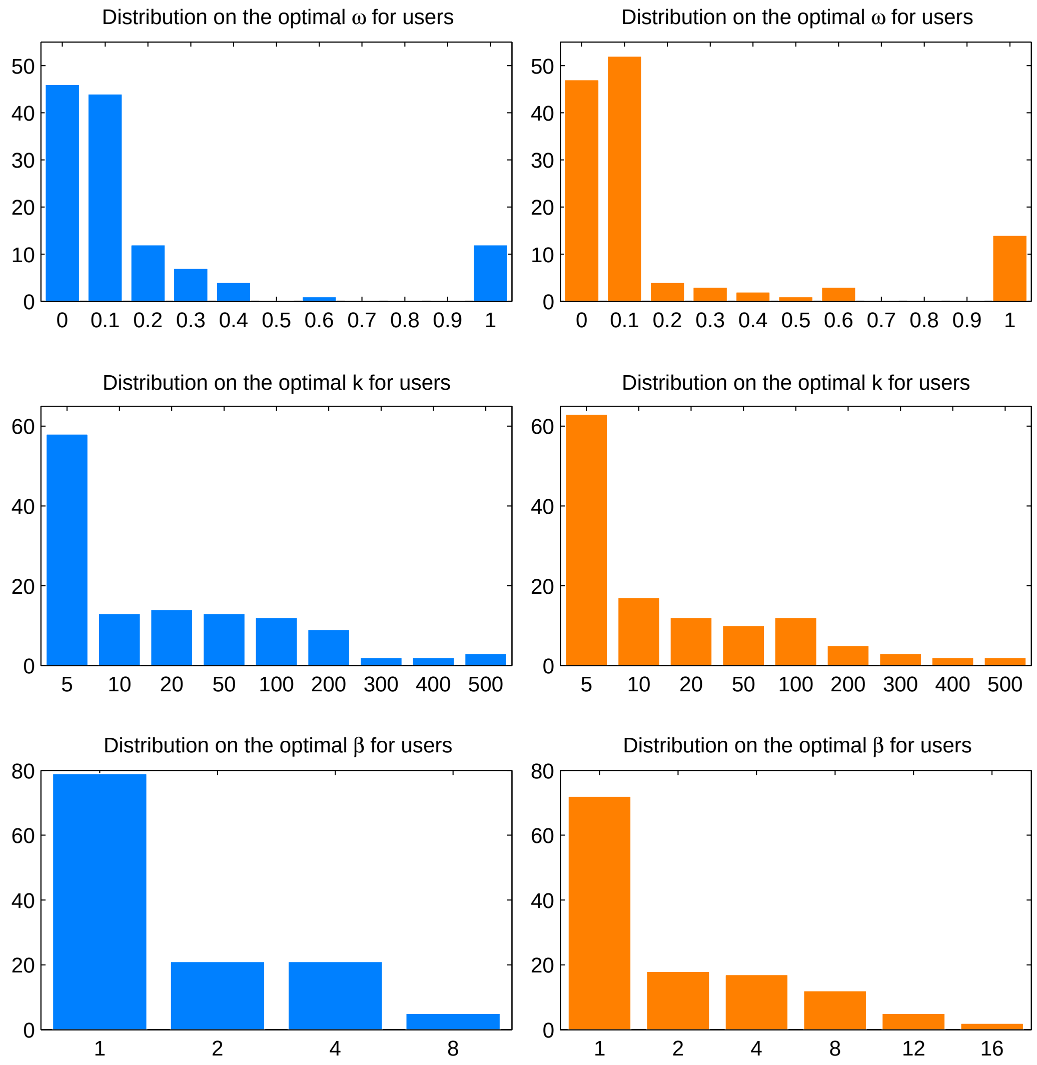
For global setting without interpolated with freq-r, trans-n1 and trans-n2 show same preference on the joint settings of β and k. This holds for both JS-divergence and L1-norm metrics. On the whole, L1-norm favors higher β and smaller k, whereas JS-divergence favors lower β and larger k.

For global setting when interpolated with freq-r, both the interpolated trans-n1 and trans-n2 favor β larger than their non-interpolated counterparts. When k is constant, larger β concentrates more weights on the nearer neighbors than those that are more distant. The increased β is larger for trans-n1 than for trans-n2. When JS-divergence is used, the interpolated trans-n1 and trans-n2 also favor larger k. In other words, more neighbors are leveraged for borrowing translations. In addition, the interpolated trans-n1 prefers higher ω than the interpolated trans-n2.

For global setting on knn methods, knn-ur and knn-ut favor the same k setting for selecting the number of nearest neighbors. Interpolating freq-r reduces the favored k in both methods, suggesting that fewer neighbors are needed when the frequency of tags is available.

The interpolated trans-u1 and trans-u2 favor the same setting on ω. They also do show competitive performance in Figure 4.3.

For individual setting on the interpolated trans-n2, we observe that small β and small k are chosen for the optimal individual performance, as shown in



4.5.a: trans-n2 using JS-divergence

4.5.b: trans-n2 using L1-norm

Figure 4.5: Distribution of Individual Settings on the Validation Set

Figure 4.5. This indicates that, users concentrate on small neighborhood and the nearer neighbors are not assigned much higher importance than the more distant neighbors, although nearer neighbors are preferred than more distant neighbors. When the optimal β and k are set, the interpolated trans-n2 depend less on freq-r. This is observed as the more users found at $\omega = 0.0$ and $\omega = 0.1$. This suggests that, using trans-n2 alone, when the optimal β and k are chosen, it is able to recommend tags that are relevant to both the query user and the query resource, with little help from freq-r. However, there are users who favor freq-r exclusively. This is observed as the relatively more users found at $\omega = 1$. It suggests that there are users who mostly follow the general crowd in assigning

tags and prefer less personalization. This observation is consistent with Sen *et al.* [99] on MovieLens[5] that when presented with prior tags (assigned by others) for the resource, users have certain tendency of adopting these tags, also known as *social influence*.

4.5 Summary

In this study, we proposed a probabilistic framework for solving the personalized tag recommendation task. This task addresses tag selection in social tagging space. Our main contribution includes leveraging the implicit social links between users that are extracted from their tagging history to find suitable tags for recommendation. Based on the approach of personomy translation, which translates the resource tags to personal tags to the query user, we proposed to leverage translations from like-minded users (neighbors). By doing so, we are able to enlarge the set of candidate tags beyond the existing vocabularies of the query resource and the query user. We found ample improvement in the recommendation performance when the set of candidate tag is enlarged.

We also noted that, while some users are more likely to use personal tags when annotating resources (known as *personal preference*), there are some other users who are more likely to follow the general crowd (known as *social influence*). For users having stronger personal preferences than social influence, we noted that, while their optimal parameter settings may include a broader neighborhood, *i.e.* $k = 100$, nearer neighbors have larger impact than the more distant neighbors, as β favors small values.

In this study, we have focused on the perspective of users. We started from the intuition that individual's tagging preference require personalized tag recommendation. However, it is also possible that the difficulty in tag recommendation may be due to the characteristics of the resources. Future research continuing this work may exploit from the perspective of resources.

[5] `www.movielens.com`

Chapter 5

Trend Discovery using Social Annotations

5.1 Introduction

Social annotations are auxiliary information users create for resources on the Web. Specifically for the scientific literature, both social tags and citing documents are social annotations to the published work. When there is an increasing attention given to a topic or an individual work, it shows up in these social annotations. In this work, we propose the task of trend discovery using social annotations, focusing on scientific publications.

Discovering and analyzing trends using social annotations for scientific publications has several useful applications. In library science and information studies, profiling the publications to support better search and reference is an important task. While the content of a publication becomes immutable once it is published, the impact it has on subsequent work can be observed over a period of time. Such impact can be shown in the social annotations, since these annotations provide temporal and topical relevance from the perspectives of the annotators. For information seekers, especially junior researchers who want to survey unfamiliar research areas, selecting interesting publications among a large collection is a challenging task. Given a selected publication, one may want to ask: *How much interest do people have on this work*? *When did such interest emerge*? *How fast was the emergence*? One may further pinpoint a particular research topic or community, *e.g. When did the interest on this work emerge from wireless networks research*?

Traditional approach to determining the impact of the published work mainly relies on citation indexes, known as *bibliometrics*. However, most citation indexes provide only a snapshot view of the citation database. Although there has been previous studies on bibliometrics with timestamps, such as *citation count* or *journal impact factor*, these studies are confined to visualizing bibliometrics at different times [19, 75]. Other than visualizing the traditional bibliometrics over time, deeper analysis on the annotation content are needed to unveil the topical impact of the published work. There has been previous studies that leverage on topic modeling to analyze the content of citing documents [75]. However, the proposed impact measure remained at the topic level, and did not include trend analysis on individual publications.

In this work, we make use of the temporal information in the social annotations to construct *social annotation profiles* for the annotated work. Based on each social annotation profile, we derive the corresponding time series, on which *trend estimation* can be performed to discover *emerging trends*. The estimated trends then allow us to compare and select interesting trends based on *how much*, *when* and *how fast* they emerge. Furthermore, we perform content analysis, through topic modeling, on the annotation content to decapsulate the multitude of impact shown in the social annotation profiles.

As noted in Chapter 2, trend discovery is a task different from event detection or burst detection. Event detection focuses on significant topic changes in a stream of documents, while burst detection detects abrupt changes in frequency in the arrival of data. In our study, an emerging trend may demonstrate gradual change in its social annotation time series.

In this study, we seek to answer the following research questions:

(i) How to find emerging trends from social annotations?

(ii) How to compare emerging trends to answer questions that are useful to researchers and information seekers?

We summarize our contributions in this research as follows.

- We use social annotations to profile scientific publications for trend discovery. Such social annotation profiles provide temporal dynamics to the annotated work from the perspectives of the annotators. We derive time series from these social annotation profiles for trend estimation.

- We propose to use sigmoid function as the trend estimator to model the social annotation time series. Such an estimator allows us to find and parameterize emerging trends, capturing characteristics such as *how much*, *when* and *how fast* the trends emerge. These characteristics provide us with useful metrics for comparing emerging trends and selecting interesting trends.

- We study three types of social annotation profiles, namely the *item-wise annotation profiles*, the *corpus-wise topic profiles* and the *item-wise topic profiles*. We examine the corpus-wise topic profiles to gain an overview of the emerging research topics. We also examine publication-specific topic profiles, seeking to understand the topical impact of individual publications as well as the important publications for a given topic. We perform topic modeling on the annotation content to construct the corpus-wise and item-wise topic profiles.

- We conduct empirical experiments using data from CiteULike (for social tags) and ACM Digital Library (for citing documents). On one hand, we perform trend analysis tasks to identify the emerging topic trends, focusing on using citing documents to the annotated work. On the other hand, we compare the emerging trends found using these two types of social annotations to commonly annotated work.

5.2 A Trend Discovery Process

An overview of our proposed trend discovery process is depicted in Figure 5.1. In order to perform trend analysis tasks that address publication-specific and topic-specific questions, we decompose the trend discovery process into three main modules, namely *topic modeling*, *trend estimation* and *trend selection and ranking*.

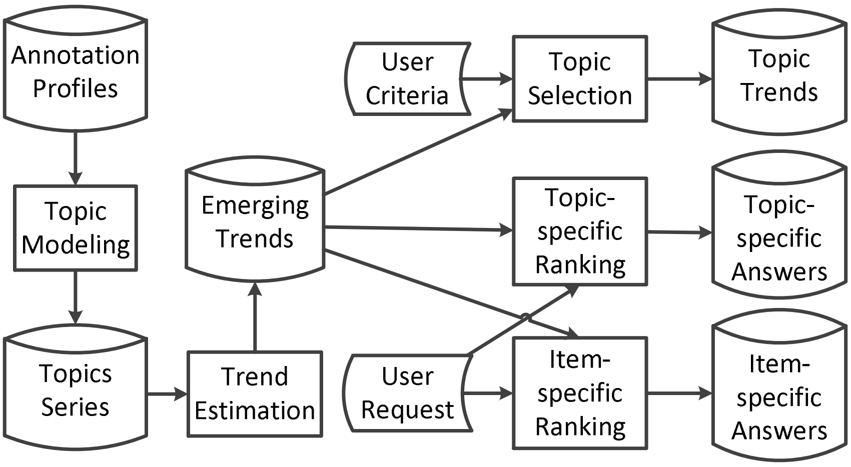


Figure 5.1: An Overview of Trend Discovery using Social Annotations

The *topic modeling* module performs content analysis on the social annotations. Social annotations for the same annotated work may come from different topics of interest. By analyzing the annotation content, we are able to decapsulate the multitude of interest. This allows us to perform trend analysis tasks that address topic-specific questions, such as *How much interest does the wireless networks research community have on the annotated work*?

The *trend estimation* module finds and parameterizes the emerging trends shown in the social annotations. To perform trend estimation, we first construct temporal profiles using the social annotations, and then derive time series corresponding to the temporal profiles. Given each time series, we perform function fitting to find the estimated trend. The trend estimator (function) should allow us to capture characteristics such as *how much*, *when* and *how fast* the trend emerges.

The *trend selection and ranking* module identifies interesting and significant emerging trends using the estimated trend parameters. To demonstrate the usefulness of the emerging trends found, we perform various topic-specific and publication-specific trend analysis tasks.

In what follows, we focus on discussing the *trend estimation* and *trend selection and ranking* modules. We leave out the details about topic modeling in this chapter, since they are discussed in Chapter 2 and Chapter 3.

5.2.1 Constructing Social Annotation Profiles

A *social annotation profile* consists of a stream of *annotation documents*. From the two types of social annotation communities, namely the social tagging community and the scientific research community, two types of social annotation documents are observed. In the social tagging community, each annotation document corresponds to one bookmark, which contains a set of tags assigned to the annotated work and a timestamp. In the scientific research community, each annotation document corresponds to one citing document, which contains the content words in citing document and a timestamp, *i.e.* the publication year. By aligning a collection of annotation documents with their corresponding timestamps, we build up a stream of annotation documents, which we call the *social annotation profile*.

We now define some terms and notations for formally representing publications and their social annotation profiles. Table 5.1 summarizes the symbols used in our discussions. We use the term *item*, denoted as i, to refer to a publication being annotated. We use the term *topic*, denoted as k, to refer to a (latent) research community specializing in an area of interest. We use the symbol $\mathbb{D}$ to denote a social annotation profile. In this study, we focus on the

Table 5.1: Notations for Social Annotation Profiles

Symbol	Definition
i	an item being annotated, *i.e.* a resource
d	a social annotation document, *e.g.* a social bookmark or a citing document
$\vec{w}_d$	the content words in the annotation document d *e.g.* the bag of tag terms in d for social tagging or the bag of words in d for citing documents
s_d	the timestamp of the annotation document d
k	a latent topic
t	a time window
$\mathbb{D}$	a social annotation profile
$\mathbb{Q}$	a time series constructed from an annotation profile
$\hat{Q}$	an estimated trend for a time series
λ	the emergence amplitude of an emerging trend
Δ	the ruling gradient of an emerging trend
τ	the emergence time (when ruling gradient is reached) of an emerging trend

following three types of social annotation profiles.

- *Item-wise document profile*, denoted as $\mathbb{D}_i$, consists of the stream of annotation documents that are used to annotate item i.
- *Corpus-wise topic profile*, denoted as $\mathbb{D}^k$, consists of the stream of annotation documents that are associated with topic k.
- *Item-wise topic profile*, denoted as $\mathbb{D}_i^k$, consists of the stream annotation documents that are associated with topic k and are used to annotate item i.

Our definition for topics follows Blei *et al.* [12]. Given a corpus consisting of a set of annotation documents, we assume that there are K topics in the corpus, *i.e.* $k \in [1, K]$. Each topic has a mixture of words, where each word has a probability of being generated by the topic, denoted by $p(w|k)$. Each annotation document has a mixture of topics, where each topic has a probability of being associated with the document, denoted by $p(k|d)$. We learn the association of each annotation document with topics by performing topic modeling on the annotation corpus.

For each social annotation profile $\mathbb{D}$, we construct the corresponding time series $\mathbb{Q} = \{(t, q_t) : 1 \leq t \leq T\}$, where t denotes a time window and q_t denotes the number of annotation documents at time window t in the social annotation profile $\mathbb{D}$. We use calendar months and publication years as time windows for social tags and citing documents respectively. Note that, each time series $\mathbb{Q}$ always shares the same superscript and/or subscript with its corresponding social annotation profile $\mathbb{D}$, *i.e.* $\mathbb{Q}_i$ for $\mathbb{D}_i$, $\mathbb{Q}^k$ for $\mathbb{D}^k$, and $\mathbb{Q}_i^k$ for $\mathbb{D}_i^k$. Without loss of generality, we omit their superscripts and subscripts in the following discussion.

To define $\mathbb{D}$ and $\mathbb{Q}$, we use d to denote an annotation document, which consists of its *annotation content* (denoted by $\vec{w}_d$) and a *timestamp* (denoted by s_d), and s_t to denote the starting timestamp of the time window t. Formally,

$$\begin{aligned}\mathbb{D} &= \left\{d_n : n \in \mathbb{N}, s_{d_n} \leq s_{d_{n+1}}\right\} \\ \mathbb{Q} &= \left\{(t, q_t) : 1 \leq t \leq T, q_t = \textstyle\sum_{d \in \mathbb{D}} I\left(s_t \leq s_d < s_{t+1}\right), q_t > 0\right\}\end{aligned}$$

where $I\left(*\right)$ is the indicator function that returns 1 if the condition $*$ is true, and 0 otherwise.

Although our current study focuses on scientific publications, the proposed model is generally applicable to other forms of Web objects with social annotations. For example, in Digg[1], the items are articles on the Web, and their social annotation profiles can be the votes (*i.e.* thumbups) from users to these articles, and the corresponding time series can be the *daily* number of votes. In Google Trends[2], the items are query keywords, and their annotation profiles can be the online news articles containing these keywords, and the corresponding time series can be the *weekly* number of these containing articles.

[1] digg.com

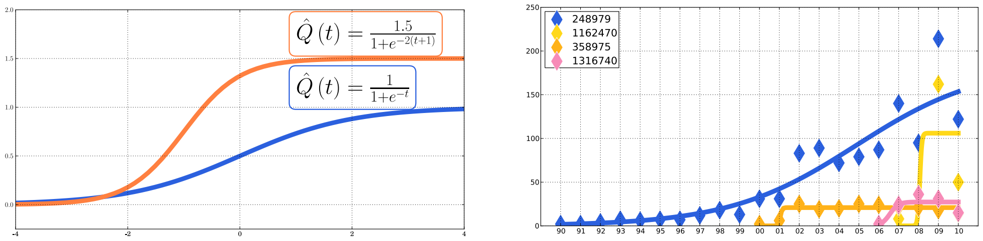
[2] www.google.com/trends

5.2.2 Estimating Trend from Time Series

For each time series derived from a social annotation profile, we apply function fitting to obtain its estimated *trend*, denoted as $\hat{Q}(t)$. Given a time series, we are interested in *how much*, *when*, and *how fast* a trend emerges, if there is any. Based on these three requirements, we choose the sigmoid function as our trend estimator. It is defined with three parameters in Eq. 5.1.

$$\hat{Q}(t) = \frac{\lambda}{1+e^{-\sigma(t-\tau)}} \qquad \text{(Eq. 5.1)}$$

Parameter λ represents the asymptotic amplitude of the curve as time goes to infinity. Parameter τ indicates the time at which the series reaches half of the asymptotic amplitude, *i.e.* $\hat{Q}(\tau) = \frac{\lambda}{2}$. It is also the time at which the curve has its largest gradient. Parameter σ controls how fast the curve approaches its asymptote. The higher σ, the faster it approaches. When the parameters are set as $\lambda = 1$, $\tau = 0$ and $\sigma = 1$, $\hat{Q}(t)$ degenerates into the standard logistic function, shown in Figure 5.2.a.



5.2.a: Standard and Parameterized Sigmoid Functions

5.2.b: Example Time Series with Fitted Sigmoid Functions

Figure 5.2: Sigmoid Functions and Fitting Examples

The choice of sigmoid function also matches with our observation from the data at hand. When plotting the $\mathbb{Q}_i$ time series for items and $\mathbb{Q}^k$ time series for topics, we see a vivid S shape, where there is a phase with low values, followed by a transition phase from low to high values, and lastly a phase of plateau, in which values remain high and do not drop much. Figure 5.2.b shows four examples of $\mathbb{Q}_i$ time series, which correspond to the citing document profiles for four publications in ACM Digital Library. It also plots the estimated sigmoid functions fitted to these time series. We observe that these time series exhibit different amplitudes, emergence times and gradients. All of these characteristics are captured by the proposed sigmoid estimator.

There exists a number of candidate functions exhibiting an S shape. For example, the exponential basis function ($\hat{Q}\left(t\right) = \frac{1-e^{-at}}{a}$) is often used for modeling financial data series [54]. However, this function assumes fastest transition at the beginning of the series. Unfortunately, this assumption is too stringent and does not apply to all data series. Another example is the Gompertz function ($\hat{Q}\left(t\right) = ae^{be^{c(t-\tau)}}$). While also showing an S shape, this function assumes asymmetry in the S shape with respect to τ, hence, it requires more parameters to control the shape of the curve. Based on empirical explorations, we choose logistic sigmoid function, for it captures the three key characteristics of emerging trends, yet makes the most general assumption about the particular shapes of the curves.

Not all time series have emerging trends. We observe the following three cases where the corresponding time series cannot find any emerging trend.

(i) The series does not fit any sigmoid curve. This happens when the trend estimator cannot find the set of suitable parameters for the series or the sum-of-squares error in the resulting fit is too large[3].

(ii) The series fits a sigmoid curve, but the estimated transition is not visible within the time range of the series. This happens when the estimated τ falls beyond the time range of the series.

(iii) The series fits a sigmoid curve with visible transition, but the transition is downward. This happens when the estimated σ is negative. The proposed sigmoid estimator is capable of capturing such downward trend.

[3] At present, we do not set a threshold on this error. We adopt the default setting of the optimizer for determining goodness of fit.

However, since downward trends are of less interest than upward trends, we do not focus on downward trends in this study.

By excluding the above three cases, we define a data series as having an *emerging trend* if it has fitted an upward sigmoid curve with the upward transition shown within its time range. In other words, a trend is *emerging* if its fitted curve satisfies both $\tau \in [1, T]$ and $\sigma > 0$.

5.2.3 Interpreting Emerging Trend Parameters

Given a time series with an estimated sigmoid curve satisfying an emerging trend, we interpret the three parameters defining the sigmoid curve as follows.

We interpret parameter λ as the *amplitude* of the emerging trend. It characterizes *how much* the trend emerges. We interpret parameter τ as the *emergence time* of the emerging trend. It characterizes *when* the trend emerges. We interpret the gradient Δ at $t = \tau$ as the *ruling gradient* of the emerging trend. It is derived as $\Delta_{t=\tau} = \frac{(\lambda\sigma)}{4}$. It characterizes *how fast* the trend emerges.

5.3 Dataset and Experimental Settings

In this section, we evaluate our proposed trend estimation model. We show how our model can be used in the following trend analysis tasks, which can potentially help the researchers and information seekers understand the different research specialties as well as the individual publications.

(i) *Discovering emerging topic trends* (Section 5.4.1). For this task, we use the corpus-wise topic profiles to find emerging topic trends. We further compare the topic trends derived from the citations and those from tagging annotations to understand the different interests of the two social annotation communities.

(ii) *Selecting important publications for a given citing topic* (Section 5.4.2); and,

(iii) *Understanding the topical impact of a given publication* (Section 5.4.3). To address these topic-specific and item-specific tasks, we compare and rank the emerging trends derived from the item-wise topic profiles.

(iv) *Identifying the most influential papers for a given conference* (Section 5.4.4). We propose trend-based metrics to select the influential papers. For evaluation, we take the ACM ICSE (International Conference on Software Engineering) conference proceedings papers as a case study.

Due to data sparseness in the social tags dataset, we conducted the experiments for task 1 on both the tagging and citation datasets. The experiments for tasks 2, 3 and 4 were conducted on the citation dataset only.

5.3.1 Data Collections

Our two data sources are CiteULike[4] (for tagging annotations) and ACM Digital Library[5] (for citation annotations). In this section, we provide an overview of the two datasets and an overlapping subset of items we identified from both data sources.

Our data dump from CiteULike is dated on May 19, 2010. It contains bookmark records to 2,419,452 items, by 49,509 users with 10,577,486 tag assignments. The bookmarks were posted between 2004 and 2010. The mostly annotated item received 879 bookmarks (*i.e.* users) with 1,455 distinct tags. The most active annotation user contributed 32,074 bookmarks (*i.e.* items) with 2,293 distinct tags.

Our data dump from ACM DL is dated on November 14, 2010. It contains 1,634,599 publication records, covering 14 types of publications. The earliest publication was published in 1956, and the latest in 2011[6]. After extracting

[4] `www.citeulike.org`

[5] `portal.acm.org`

[6] Since publications with recorded publication year of 2011 are minorities, we excluded them and kept those whose publication year is no later than 2010.

the citation links among these publications, we identified 495,190 publications citing others and 549,098 publications being cited.

The Joint Set of Items

Our task 1 of discovering emerging topic trends is concerned with publications having both tagging and citation annotations. However, the publication collections covered by CiteULike and the ACM DL are not identical, although there exist overlaps between them. We therefore seek publications that have both tagging annotations in CiteULike and citation annotations in ACM DL. Fortunately, CiteULike provides linkout data from items in CiteULike to other digital libraries. The linkout data we obtained, dated on December 9, 2010, contains 6,311,250 linkout records, in which 66,388 items are linked to ACM DL . We noticed that multiple CiteULike items linking to the same ACM DL publication. After resolving co-references in these links, we identified 64,066 distinct publications in ACM DL. Having extracted annotations for these publications from both sources, 44,123 publications are identified as having both tagging and citation annotations. In what follows, we refer to publications in this subset as *items*. The statistics of the extracted data are shown in Table 5.2.

Table 5.2: Statistics of Items in ACM DL, CiteULike and the Joint Set

Statistics	ACM DL	CiteULike
Number of distinct items	64066	66388
Number of items having annotations	51936	52223
Number of items having annotations in both		**44123**

For all items in the joint set, we construct the annotation profiles from both the citation and tagging data sources. All annotation documents contained in these profiles constitute our annotation corpora for citations and tags respectively.

5.3.2 Topic Modeling on the Datasets

We compiled a topic learning corpus consisting of the content of all items in the joint set and the content of all citing documents for these items in ACM DL . Specifically, for the 44,123 items in the joint set, 327,857 ACM DL records (cum documents), including the cited and the citing documents, are included for topic learning.

For each record in the corpus, we concatenate its title and abstract to form the document content. Stopwords and words appearing in fewer than 5 documents are removed. Consequently, documents with fewer than 5 valid word tokens are also removed. As a result, 313,268 documents containing 68,725 distinct words are used for topic learning.

We adopt the GibbsLDA++[7] software tool for learning topics from the corpus. Following the settings in [75], we also set the total number of topics to 200, *i.e.* $K = 200$. Given the topic learning result, we associate a document with a topic if more than 10% word tokens in the document are assigned to the particular topic [75]. The choice of 10% is to filter out minor topics assigned to the documents by chance. As a result, each document is associated with 2.03 topics on average.

The learned topic model is further used for learning topic associations for the tagging annotation corpus, which consists of all annotation documents assigned to items in the joint subset.

5.4 Experimental Results

5.4.1 Topic Trends for Annotation Corpora

In this section, we present the topic trends found in the two annotation corpora. We seek to compare the emerging topic trends by answering the following questions:

[7] gibbslda.sourceforge.net

- *What are the topics that emerge mostly in each annotation community?*
- *What are the topics that emerge fastest in each annotation community?*
- *What are the topics that emerge most (or least) recently in each annotation community?*

To answer these questions, we examine the corpus-wise topic profiles $\mathbb{D}^k$ and the corresponding time series $\mathbb{Q}^k$. We apply the sigmoid estimator to find emerging topic trends and compare the trends by their emergence amplitude (λ^k), ruling gradient (Δ^k) and emergence time τ^k, as defined in Section 5.2.3. We discuss the different trends found in the citation *vs.* those in the tagging communities.

Topic Trends in the Citation Community

Tables 5.3, 5.4 and 5.5 list the top 10 topics in the citation annotation corpus with the highest emergence amplitude, largest ruling gradient and most recent (5.5(a)) and least recent (5.5(b)) emergence time respectively.

Table 5.3: Topics in Citation Community with Highest Amplitude

λ^k	Topic	Top Keywords
1273.5	155	channel channels capacity interference spectrum power
1260.4	160	image images segmentation color regions region method
1141.5	145	sensor networks nodes network wireless node sensors
1081.0	189	routing networks ad hoc network nodes multicast protocol
1040.4	073	wireless networks access network throughput protocol mac
989.1	006	genetic evolutionary algorithm algorithms optimization
934.5	138	3d camera images image scene 2d reconstruction
911.2	106	key secure security signature protocol authentication
895.4	013	coding compression codes encoding binary rate error
826.1	103	user interface interaction interfaces users input interactive

The emergence amplitude λ^k indicates *how much* topic k emerges in the annotation community, estimated as of the latest time window. Topics that emerge with the largest amplitude in the citation community, shown in Table 5.3, include: topic 155 on *channel capacity*, topic 160 on *image segmenta-*

tion, topics 145 and 073 related to *wireless sensor networks* and topic 106 on *security and authentication*.

Table 5.4: Topics in Citation Community with Largest Ruling Gradient

Δ^k	Topic	Top Keywords
6474.8	155	channel channels capacity interference spectrum power
262.4	145	sensor networks nodes network wireless node sensors
223.0	166	number asynchronous show strong consensus synchronous
172.6	073	wireless networks access network throughput protocol mac
168.1	184	medical diagnosis health patients clinical care using
152.7	160	image images segmentation color regions region method
143.1	135	face recognition fusion facial using expressions features
136.4	157	social community online communities users email people
123.5	189	routing networks ad hoc network nodes multicast protocol
122.3	130	security attacks attack secure malicious authentication

The ruling gradient Δ^k indicates *how fast* topic k emerges in the annotation community. Topics emerge fastest in the citation community, shown in Table 5.4, include: topic 155 on *channel capacity*, topics 145 and 073 related to *wireless sensor networks*, topics 160 and 135 related to *computer vision* and topic 157 on *social community*.

By comparing the topics listed in Table 5.3 and those in Table 5.4, we note that, although topic 157 (on *social community*) is not among the most popular topics (*i.e.* high emergence amplitude), it has shown very intense growth of interest (*i.e.* large ruling gradient) in the citation community. In contrast, topic 155 is both popular and has most intense growth of interest.

The emergence time τ^k indicates the time at which the most intense emergence of topic k is observed. Table 5.5(a) shows the topics with the most recent τ^k in the citation community, and Table 5.5(b) shows the topics with the least recent τ^k. Topics showing intense emergence most recently include: topic 013 on *coding and compression*, topic 057 on *neural prediction networks* and topic 161 on *machine learning*. We also note topic 020, which does not seem to represent a meaningful topic.

Topics that show intense emergence least recently include: topic 173 on *rendering techniques for computer graphics*, topic 185 on *hardware circuits de-*

Table 5.5: Topics in Citation Community Ranked by Emergence Time

(a) Top 10 Topics with Most Recent Emergence

τ^k	Topic	Top Keywords
2009.3	013	coding compression codes encoding binary rate error
2009.3	110	conditions sufficient condition given certain whether
2009.0	158	simulation simulations results using simulator realistic
2008.9	069	errors fault reliability failure failures error recovery
2008.3	064	linear matrix sparse matrices polynomial symmetric
2008.2	121	system operating hardware platform implementation
2008.2	172	estimation parameters error estimate parameter accuracy
2008.1	020	one two another hand latter ie different paper former
2008.1	076	task tasks transfer perform performing performed using
2008.0	155	channel channels capacity interference spectrum power

(b) Top 10 Topics with Least Recent Emergence

τ^k	Topic	Top Keywords
2004.0	027	retrieval information relevance terms documents term
2004.1	079	may however often many lead result even conflict occur
2004.1	107	study results experiment effects effect found participants
2004.2	125	service services qos composition providers quality web
2004.2	048	complexity bound bounds lower upper polynomial number
2004.3	169	matching match string length pattern two common pair
2004.3	083	data sets collected processing large amount warehouse raw
2004.3	053	average using 10 rate times less per 20 compared percent
2004.3	039	state hybrid continuous states markov transition discrete
2004.3	051	sequence sequences gene cell biological expression protein

sign, topic 068 on *web sites and pages* and topic 174 on *frequent pattern mining*. Unfortunately, topic 079, which is among the least recently emerging topics, appears to be a non-genre topic.

Topic Trends in the Tagging Community

For the corpus-wise topic profiles using tagging annotations, the top 10 topics with the highest emergence amplitude, largest ruling gradient and most (*vs.* least) recent emergence time are shown in Tables 5.6, 5.7 and 5.8 respectively.

We note that the top topic trends in Tables 5.6 and 5.7 are mostly related to web and text mining. These topics include topic 157 on *social community*, topic 089 on *recommender systems*, topic 027 on *information retrieval* and topic 104 on *tagging*. This observation suggests that the annotation community of CiteULike have been actively annotating publications in web and text mining

Table 5.6: Topics in Tagging Community with Highest Amplitude

λ^k	Topic	Top Keywords
124.1	122	2006 2007 2005 2008 2004 thesis 2009 acm vldb sigmod
62.7	157	social community wiki email socialnetwork blogs blog
55.2	089	recommender collaborativefiltering personalization
50.2	027	ir retrieval relevancefeedback relevance queryexpansion
44.5	103	hci interaction interface ui user usability userinterface
41.9	104	tagging folksonomy tag tags folksonomies 519 flickr
40.4	068	web hypertext www hypermedia pagerank accessibility
40.2	038	search ranking websearch google rank informationretrieval
39.1	195	informationretrieval text wikipedia summarization
38.9	167	ontology semantic annotation semanticweb taxonomy

Table 5.7: Topics in Tagging Community with Largest Ruling Gradient

Δ^k	Topic	Top Keywords
35.1	122	2006 2007 2005 2008 2004 thesis 2009 acm vldb sigmod
29.6	027	ir retrieval relevancefeedback relevance queryexpansion
24.3	148	p2p network networks peertopeer dht overlay topology
23.9	068	web hypertext www hypermedia pagerank accessibility
20.4	189	routing manet adhoc sensornetworks multicast dtn mobil
20.4	082	collaboration cscw collaborative awareness supported work
19.1	007	mobile ubicomp pervasive ubiquitous mobility computing
18.9	157	social community wiki email socialnetwork blogs blog
18.6	089	recommender collaborativefiltering personalization
16.3	104	tagging folksonomy tag tags folksonomies 519 flickr

related research. In contrast, users from other research specialty have been less active in CiteULike.

The time range in the tagging annotation corpus is drastically different from that in the citation annotation corpus. The emergence time shown in Table 5.8 indicates the time at which the tagging community in CiteULike demonstrates the most intense surge of interest in tagging the topics. We note that topic 134 on *arithmetic computing* shows the most recent surge, while topic 147 on *digital libraries and metadata* surges the earliest.

Note that the topic mixtures have changed after learning on the tags. Many abbreviations now have higher probabilities of being generated by the topics. For example, the tag ir in topic 027 and the tag hci in topic 103, shown in Table 5.6.

Table 5.8: Topics in Tagging Community Ranked by Emergence Time

(a) Top 10 Topics with Most Recent Emergence

τ^k	Topic	Top Keywords
2009.5	134	computation floatingpoint fast computing fixedpoint
2009.5	129	efficiency redundant redundancy naturallanguageproce
2009.5	077	energy power consumption lowpower efficiency energyeff
2008.6	099	log isi iui time ni datastructure diameter computing
2008.3	093	java compiler staticanalysis compilers program debugging
2008.0	025	svm kernel vector support machine regression kernels
2007.7	084	structure qa questionanswering structured structures
2007.7	141	parallel gpu mapreduce gpgpu multicore parallelism
2007.7	122	2006 2007 2005 2008 2004 thesis 2009 acm vldb sigmod
2007.6	024	characterization family definition classic generalization

(b) Top 10 Topics with Least Recent Emergence

τ^k	Topic	Top Keywords
2004.8	020	ie hand variants humanfactors anomalydetection
2004.9	117	contribution machinelearning second longtail
2005.0	047	problems classic complexity problem solutions algorithmic
2005.1	043	input function output operator fp bent functions
2005.1	154	video multimedia streaming media stream streams trecvid
2005.2	118	design methodology usercentered ucd prototyping
2005.2	048	complexity dnf informationtheory seminal combinatorics
2005.2	182	knowledge knowledgemanagement expertise expert
2005.2	164	km enterprise management ict organization organizations
2005.3	147	metadata digital library book hardcopy content browsing

5.4.2 Influential Items for Topics

Given a topic, which are the influential publications? To answer this question, we examine the topic trends estimated from the item-wise topic profiles $\mathbb{D}_i^k$. In particular, for a topic k, we are interested in items with the largest λ_i^k. The parameter λ_i^k indicates the popularity of the item i being annotated for the topic k. We select two topics noted previously in the citation corpus, namely topic 155 and topic 157, as case studies.

For each topic, we show its corpus-wise emergence time τ^k and its top keywords learned from the citation corpus. We rank the items for each topic by the emergence amplitude λ_i^k of their item-wise topic trends. For comparison, we also show the corresponding emergence time τ_i^k, ruling gradient Δ_i^k and the citation count statistics cc_i in ACM DL. Lastly, we show the item ID, title and publication type in recorded in ACM DL, where WB denotes whole book, JA denotes journal articles and PP denotes conference proceeding papers.

Table 5.9: Top Items for Topic 155

Topic	τ^k	Top Keywords				
155	2008.0	channel channels capacity interference spectrum power				
λ_i^k	τ_i^k	Δ_i^k	cc_i	Item	Title	Type
200.0	2008.1	1018.6	2410	129837	Elements of information theory	WB
194.5	2008.1	1378.9	1239	993483	Convex Optimization	WB
146.5	2008.1	782.5	487	609324	On Limits of Wireless Communications in a Fading Environment when Using Multiple Antennas	JA
93.0	2008.1	562.5	242	1162470	NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey	JA
43.5	2008.1	224.2	1121	248979	Matrix computations (3rd ed.)	WB
25.0	2008.1	143.8	76	1161129	Hot topic: physical-layer network coding	PP
15.0	2008.1	73.9	57	1282425	Embracing wireless interference	JA
13.0	**2007.2**	33.9	114	1148681	Raptor codes	JA
12.0	2008.1	75.8	139	1159942	XORs in the air: practical wireless network coding	JA
12.0	2008.1	86.0	191	19572	Topics in matrix analysis	WB

Table 5.10: Top Items for Topic 157

Topic	τ^k	Top Keywords				
157	2005.6	social community online communities users email people				
λ_i^k	τ_i^k	Δ_i^k	cc_i	Item	Title	Type
14.7	2007.1	68.6	107	1240772	Why we tag: motivations for annotation in mobile and online media	PP
11.3	2006.0	60.4	101	988739	Information diffusion through blogspace	PP
11.0	2007.1	54.9	50	1242685	Analysis of topological characteristics of huge online social networking services	PP
10.3	2007.1	65.4	34	1240695	A familiar face(book): profile elements as signals in an online social network	PP
10.1	**2001.0**	51.5	196	358975	Interaction and outeraction: instant messaging in action	PP
10.0	2008.0	47.3	60	1341558	Can social bookmarking improve web search?	PP
9.5	2008.1	48.3	23	1397742	Growth of the flickr social network	PP
8.0	2007.1	38.8	61	1124885	Understanding photowork	PP
8.0	2008.1	43.4	50	1242598	Wherefore art thou r3579x?	PP
7.3	2007.0	59.1	108	1316740	Combating web spam with trustrank	PP

Table 5.9 shows the top 10 items with the largest λ_i^k for topic 155. Among these items, the top journal papers and top conference proceedings papers appear more directly addressing specific research problems in the field of network coding, while the top books appear more on fundamental theories. In Table 5.9, we also highlight the items whose item-wise topic trends emerge earlier than the corpus-wise topic trend for the corresponding topic, *i.e.* $\tau_i^k < \tau^k$. For topic 155, the journal paper *Raptor codes* qualifies this selection.

Table 5.10 lists the top 10 items with the largest λ_i^k for topic 157. Interestingly, unlike topic 155, no book is found among the top items for topic 157.

These top items all appear mostly addressing specific social Web services, and no fundamental book has been commonly adopted. For topic 157, the paper *Interaction and outeraction: instant messaging in action* shows early emergence, *i.e.* its item-wise topic trend emerges earlier than the corpus-wise topic trend for topic 157.

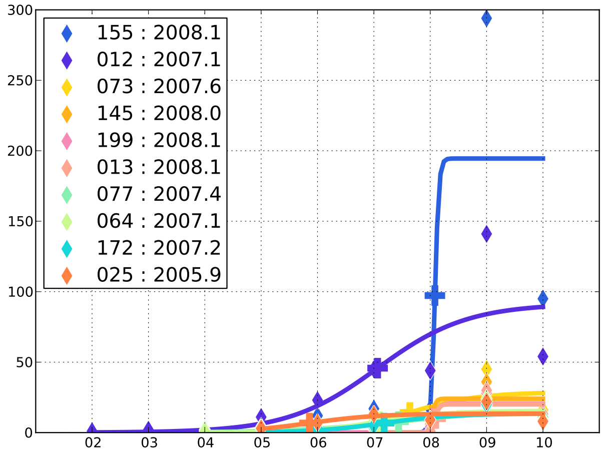
By comparison, topic 155 and topic 157 demonstrate different patterns in the top cited items. Topic 155 has theory-oriented books to serve as its intellectual bases, whereas topic 157 does not. Upon examination, we find these two patterns generally exist in the top cited items for other topics.

5.4.3 Emerging Topics for Items

In this section, we study the emerging trends for individual publications. Specifically, we examine the item-wise topic profiles $\mathbb{D}_i^k$ for a given item. Trends that emerge from the respective annotation series $\mathbb{Q}_i^k$ reveal the timeline of impact an items has made in the respective topics. We select two publications noted in the previous section, which are among the top items for the shown topics in Table 5.9 and Table 5.10.

Figure 5.3 plots the top emerging topics for the book *Convex optimization* by Boyd and Vandenberghe. Two notable, and related, topics citing this book are topic 012 on *optimization* theory, and topic 155 on *channel capacity*, which is an application domain of the theory. While the topic related to optimization theory shows a steady growth over the years, the topic on the application shows sharp and intense surge in citing this book. This observation suggests that much attention on the book comes from the applications, such as channel capacity for network coding.

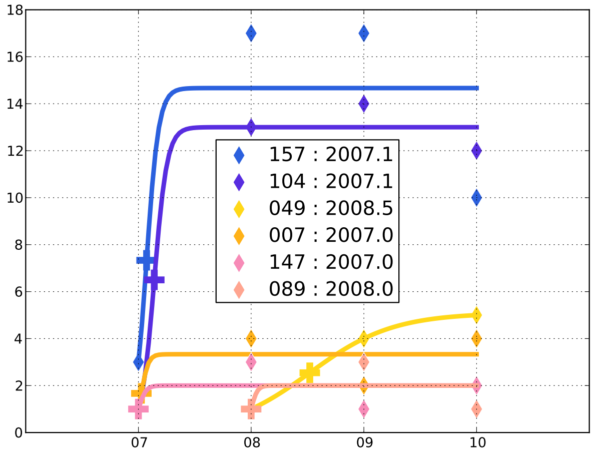
In our extended studies, we also observe similar patterns in other items, *e.g.* the book *Elements of Information Theory* by Cover and Thomas, seen in Table 5.9. In general, for emerging trends found in citing the same theory-oriented item, topics on fundamental theories show steady growth, while topics



155 : channel channels capacity interference spectrum power diversity rate
012 : optimization problem linear function optimal formulation objective
073 : wireless networks access network throughput protocol mac ieee layer
145 : sensor networks nodes network wireless node sensors energy sensing
199 : noise signal filter filtering signals filters proposed frequency noisy
013 : coding compression codes encoding binary rate error decoding
077 : power energy consumption voltage reduce scaling show leakage low
064 : linear matrix sparse matrices polynomial symmetric polynomials
172 : estimation parameters error estimate parameter accuracy estimates
025 : vector support machine kernel machines svm regression vectors using

Figure 5.3: Emerging Trends for the Book *Convex optimization*

on applications may show intense surge.



157 : social community online communities users email people personal
104 : identification using identify paper ii identifying tags used use based
049 : factors study influence use impact research perceived results findings
007 : mobile devices computing device ubiquitous smart mobility pervasive
147 : digital content library metadata paper libraries use contents creation
089 : users user preferences recommendation items profile profiles

Figure 5.4: Emerging Trends for the Paper *Why we tag*

For the conference proceedings paper *Why we tag: motivations for annotation in mobile and online media* by Ames and Naaman (the first item in Table 5.10), the top emerging topic trends for this paper are plotted in Figure 5.4. Among six topics that emerge, four shows sharp and intense emergence soon after the paper's publication. Two having large emergence amplitude are topic 157 on *social community* and topic 104 on *tag-based identification*. Two topics with lower emergence amplitude are topic 007 on *mobile devices* and 147 on *digital libraries*. Work studying *tag recommendation* (topic 089) begin to cite this paper only later with a lower volume.

5.4.4 Identifying Influential Papers for ICSE Conference

In this section, we propose to use trend-based metrics to identify influential publications. We use proceedings papers from proceedings of the ACM International Conference on Software Engineering (ICSE for short)as a case study. We demonstrate that our trend-based metrics are effective in ranking the most influential papers among their peers than pure citation count.

The ACM ICSE conference has the tradition of giving awards to papers that [90]

> is judged to have had the most influence on the theory or practice of software engineering during the 10 years since its original publication.

From 1989 to 2010, ICSE has given 22 papers the *most influential paper award*s (ICSE Award for short). In each award year, one or two awards are given. There are two years in which no award was given, *i.e.* 2000 and 2006; there are also two years in which two papers received the award, *i.e.* 1997 and 2010. Among the 22 papers receiving the award, we identified 21 papers in ACM Digital Library.

5.4.4.1 Ranking Task

We define our ranking task as follows.

- For each conference year (denoted by Y_c), we first collect the papers published in Y_c as the assessment candidates. We use N_c to denote the total number of candidates for each corresponding Y_c.
- We then collect annotation data from ACM DL for these candidates up to the assessment year (denoted by Y_a). We define Y_a to be the ICSE Award year for each corresponding Y_c.
- We compute the ranking metrics for each candidate paper based on data collected up to Y_a.
- We rank the candidate papers for each Y_c and evaluate against the ground truth at the corresponding Y_a.

We adopt a two-phase ranking strategy. In the first phase, we shortlist candidate papers for each Y_c by their citation count up to Y_a. We use M_c to denote the size of the shortlist. In the second phase, we re-rank these shortlisted candidates using the various metrics. We adopt this strategy because it best reflects the selection process for ICSE Award.

5.4.4.2 Ranking Metrics

We categorize our ranking metrics into three groups, shown as follows:

- sum.cd, which counts the total number of citing documents up to Y_a. This metric is not based on emerging trend estimation. Given that the ICSE Award first shortlists candidate papers by citation count, this ranking metric is a very strong baseline.
- Metrics based on the emerging trends estimated for $\mathbb{Q}_i$ include

  - amp.cd, the emergence amplitude (λ_i) estimated for $\mathbb{Q}_i$; This metric indicates how much interests does the research community have on the candidate paper;
  - grd.cd, the ruling gradient (Δ_i) at $t = \tau$ estimated for $\mathbb{Q}_i$; This metric indicates the maximum surge of interest in the candidate paper;
  - est.cd, the accumulated values on the estimated sigmoid curve for $\mathbb{Q}_i$. Formally, it computes the sum $\sum_t \hat{Q}_i(t)$ where $t \in [Y_c, Y_a]$ for each candidate i.

  Note that, candidates not showing emerging trend in $\mathbb{Q}_i$ will not be ranked by this group of metrics.

- Metrics based on the emerging topic trends estimated for $\mathbb{Q}_i^k$ include
  - numser, the number of emerging topic trends estimated for the candidate. Formally, it computes $\sum_k I\left(Q_i^k\right)$ where $I\left(Q_i^k\right)$ is an indicator function that returns 1 if the corresponding $\hat{Q}_i^k(t)$ is emerging and 0 otherwise;
  - ampsum, the sum of the emergence amplitude of the emerging topic trends estimated for the candidate, *i.e.* $\sum_k \lambda_i^k \times I\left(Q_i^k\right)$;
  - grdmax, the maximum ruling gradient among the emerging topic trends estimated for the candidate, *i.e.* $\max_k \Delta_i^k \times I\left(Q_i^k\right)$;

  Note that, candidates not showing emerging topic trend in $\mathbb{Q}_i^k$ for any k will not be ranked by this group of metrics. For ampsum, grdmax and estsum, the summation operator can also be replaced by the maximum operator, and vice versa. We explored both settings, and report only the results using the listed operator in this section.

To construct the $\mathbb{Q}_i$ and $\mathbb{Q}_i^k$ time series for deriving trend-based metrics, we prefix a zero at Y_c for each candidate. Formally,

$$\mathbb{Q} = \left\{(t, q_t) : Y_c \le t \le Y_a, q_t = \textstyle\sum_{d\in\mathbb{D}} I\left(s_t \le s_d < s_{t+1}\right), q_t > 0, q_{Y_c} \ge 0\right\}$$

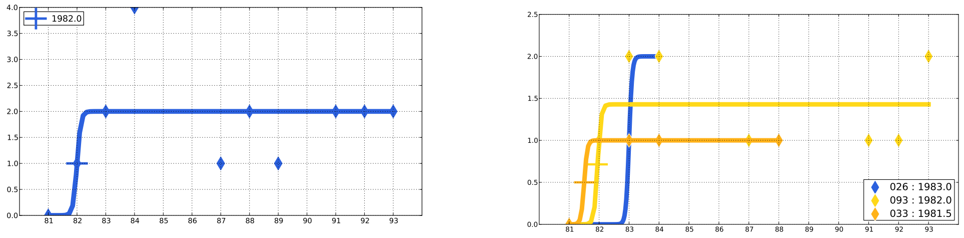
5.4.4.3 Ranking Evaluation

Table 5.11 shows the rankings of the ICSE Award-winning papers by the various metrics. We set $M_c = 10$ for all Y_c. For each paper, we show its corresponding year of publication (denoted by Y_c), year of award (denoted by Y_a), total number of candidate peers for the same year (denoted by N_c), its item ID and its cumulative citation count up to Y_a (denoted by ccc$_i$). For each ranking metric, we show the ranking position of the award-winning paper and the total number of candidates that qualify the ranking metric, *i.e.* candidates showing emerging trends. When an award-winning paper is successfully ranked ahead of all other candidate peers by a ranking metric, we highlight the corresponding cell in boldface. Note that, for $Y_c = 2000$ and $Y_c = 1987$, two papers were given the ICSE Award.

Table 5.11: Ranking ICSE Award Papers

Y_c	Y_a	N_c	Item	ccc$_i$	sum.cd	amp.cd	grd.cd	est.cd	numser	ampsum	grdsum
2000	2010	182	337234	256	**1/10**	**1/9**	3/9	**1/9**	**1/9**	**1/9**	3/9
2000	2010	182	337209	41	13/10	−/9	−/9	−/9	−/9	−/9	−/9
1999	2009	115	302457	232	**1/10**	**1/8**	3/8	**1/8**	**1/8**	**1/8**	2/8
1998	2008	68	302181	92	**1/10**	**1/4**	2/4	**1/4**	**1/4**	**1/4**	**1/4**
1997	2007	123	253236	45	**1/10**	3/8	3/8	2/8	**1/8**	2/8	2/8
1995	2005	33	225016	13	8/10	2/5	5/5	4/5	4/5	4/5	4/5
1994	2004	45	257745	89	**1/10**	−/6	−/6	−/6	−/6	−/6	−/6
1993	2003	49	257610	16	**1/10**	−/8	−/8	−/8	−/8	−/8	−/8
1992	2002	34	143098	10	9/10	5/6	3/6	5/6	−/5	−/5	−/5
1991	2001	46	256748	29	2/10	2/9	9/9	2/9	**1/9**	**1/9**	3/9
1989	1999	64	74588	45	**1/10**	−/6	−/6	−/6	−/4	−/4	−/4
1988	1998	44	55861	36	**1/10**	−/8	−/8	−/8	−/6	−/6	−/6
1987	1997	43	41767	19	7/10	−/3	−/3	−/3	−/2	−/2	−/2
1987	1997	43	41766	97	**1/10**	−/3	−/3	−/3	−/2	−/2	−/2
1985	1996	58	319624	7	5/10	7/8	3/8	6/8	**1/4**	3/4	2/4
1984	1995	75	801999	6	9/10	5/8	4/8	4/8	−/4	−/4	−/4
1982	1994	48	807765	4	8/10	6/8	3/8	7/8	−/5	−/5	−/5
1981	1993	51	802557	17	2/10	3/6	3/6	3/6	**1/5**	**1/5**	**1/5**
1979	1992	60	802918	15	3/10	5/8	4/8	4/8	5/7	4/7	7/7
1978	1991	47	803218	8	3/10	5/9	4/9	4/9	3/7	6/7	4/7
1976	1986	104	807708	6	17/10	−/4	−/4	−/4	−/4	−/4	−/4

As expected, sum.cd performs strongly. Among the 21 award-winning papers, it is able to rank them ahead of other candidate peers for 9 different Y_c. We also note that, for $Y_c = 2000$ and $Y_c = 1976$, even the award-winning papers were not among the shortlisted candidates. There are 5 years, namely $Y_c = 1994, 1993, 1989, 1988, 1987$ respectively, for which sum.cd successfully ranks the award-winning paper ahead but none of the trend-based metrics is able to. Nonetheless, there are 3 years, namely $Y_c = 1991, 1985, 1981$ respectively, for which sum.cd is not able to rank the award-winning paper at top but some trend-based metrics do. These award-winning papers include *Program slicing* by Mark Weiser (1981) and *Tolerating inconsistency* by Robert Balzer (1991). To understand why they are supported by the trend-based metrics, we plot the emerging trends estimated for these papers, shown in Figures 5.5 and 5.6.



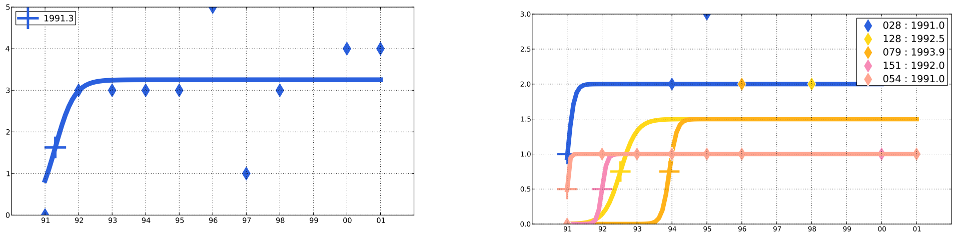
5.5.a: Trends Estimated for $\mathbb{Q}_i$

5.5.b: Trends Estimated for $\mathbb{Q}_i^k$

026 : test testing tests module cases modules coverage unit
093 : java compiler staticanalysis compilers program debugging
033 : tools tool support development environment developed integrated

Figure 5.5: Emerging Trends for the Paper *Program slicing*

Both Figures 5.5 and 5.6 show a number of emerging topic trends for the two papers, namely 3 and 5 respectively. The metric numser ranks these papers ahead of their candidate peers. This suggest that these papers attracted interest from a wide range of research communities as compared to their peers.



5.6.a: Trends Estimated for $\mathbb{Q}_i$

5.6.b: Trends Estimated for $\mathbb{Q}_i^k$

028 : process processes business workflow paper handling approach
128 : logic reasoning logics logical semantics formulas formula
079 : may however often lead many result even conflict occur
151 : requirements aspects scenarios paper aspect concerns
054 : software development engineering project projects developers

Figure 5.6: Emerging Trends for the Paper *Tolerating inconsistency*

5.5 Summary

In this research, we proposed using social annotations to profile scientific publications for trend discovery. We proposed a trend discovery process (shown in Figure 5.1) and a trend estimation method (the sigmoid estimator) for the task at hand. Leveraging on topic modeling, we derived topic series from the annotation profiles, and performed trend estimation to find emerging topic trends. With the discovered trends from the social annotations, we were able to perform analysis tasks for understanding, comparing and selecting the scientific publications.

Our empirical findings from the proposed trend analysis tasks can be summarized as the following:

- We find a number of topics that have attracted intense emergence of interest in the citation annotation corpus. These include topics on *wireless channels and sensor networks* and *social community.*

- When examining the top cited items for these topics, we observe notable differences in the most important publications for the topics. On one hand, the topic on *wireless channels and sensor networks* has been citing theory-oriented books as its intellectual bases. On the other hand, there is no commonly cited book found for the topic on *social community.* The top cited papers focus on specific problems related to different online social media.

- We have also observed different trends in citing theory-oriented books. While the core theory topics show steady growth over the years, the application topics may surge with intense emergence.

- We also demonstrated the ability to select influential papers using trend-based metrics. Our case study using the ICSE Award showed that even when pure citation count was not able to rank the award winning paper ahead of their candidate peers, our trend-based metrics support these papers for their number of emerging topic trends found.

Based on the proposed trend discovery process, future research can explore more interesting trend analysis tasks for understanding the scientific literature. For example, one may be interested in finding *co-emerging* topics, *e.g.* two or more topics show strong correlation in citing the same or similar work, or in finding *late-emerging* topics, *e.g.* a topic was not seen impacted by some work published earlier, but as time goes by, these work begin to have impact on the topic. How to formally define and solve these tasks are the steps forwards. Our proposed trend discovery process provides quantitative measures that can potentially benefit these tasks.

Chapter 6

Conclusion

6.1 Concluding Remarks

As social media continues to grow, users are equipped with more and more interactive means for contributing content to the Web and learning from the Web. Social tagging, as one of the most popular activities on social media, has created an information space for navigating vast amount of content on the Web. Navigation in this information space relies fundamentally on the links between information objects. To meet the complex information needs of users, more tools are demanded to select the information objects. This dissertation focused on studying tasks that facilitate navigation in social tagging systems. To address link sparseness, we study the task of tag prediction to increase the navigability of the resources. To address object selection, we study the task of personalized tag recommendation and propose new metrics based on the temporal profiles of the resources, helping users navigate this information space effectively.

In Chapter 3, we studied the tag prediction task, which aims to predict tags for the untagged or inadequately tagged resources. This task addresses link sparseness between resources and tags. The task of tag prediction is challenging on its own merit. The novelty of our approach includes the use of a probabilistic topic model which provides probabilistic interpretations on the semantic relationships between resources and tags via topics. By hypothesizing that *topics that are discussed more often in the document are likely to have more tags corresponding to these topics*, we proposed `LDAtgg` model for predicting tags for Web pages. We formulated the solution into two phrases, namely training and prediction. We developed a Gibbs sampling algorithm for training the model parameters. We adopted Bayesian inference for estimating the probabilities of candidate tags for the test pages.

Our experiments conducted using a novel collection of news articles showed promising performance in the tag prediction accuracy. Our `LDAtgg` model using 100 topics outperforms the strongest baseline method by over 20% for the top 5 predictions. Upon further analysis on the prediction errors, we noted two major challenges of the current approach. First, evaluation on the prediction accuracy is non-trivial. This is partly due to the varying adequacy of the ground truth tags for the test documents, and partly due to the morphological mismatches between the ground truth tags and the predicted tags. Second, `LDAtgg` model favors frequent tags in the tag vocabulary. In other words, tags seen more frequently in the training documents are more likely to be predicted for the test documents. Since frequent tags are likely to represent the coarse grain semantics, *e.g.* `tabletPC` has a coarser grain semantics than `iPad2`, this characteristic of the model has two folds. On one hand, the predicted links between the resources and the coarser grain tags make the resource navigable from other resources more easily. On the other hand, if limited to the top few, these tags may not represent the unique semantics of the resources.

In Chapter 4, we studied the personalized tag recommendation task. The objectives of personalized tag recommendation are twofold. On one hand, the recommended tags should be relevant to the particular resource. On the other hand, the recommended tags should also be consistent with the personal tagging preferences of the query user. This task addresses link selection on tags for a given pair of resource and user. The need for personalization distinguishes this task from tag prediction. We proposed a probabilistic framework that leverages the implicit social links between users, *i.e.* like-minded users who show similar tagging preference patterns, to find candidate tags for recommendation. We addressed personalization by modeling the probabilistic tagging preference patterns of individual users, *i.e.* the probabilities of translating resource tags to personal tags. We devised distributional similarity measures for comparing such tagging preference patterns between users.

Our experiments conducted on a benchmark dataset from BibSonomy suggested that leveraging like-minded users had clear advantage over their target-user-solely counterparts in terms of recommendation accuracy. Based on parameters tuned for individual users across a range of similarity measures and neighborhood sizes, we discussed the degree of personal preferences *vs.* social influences shown by users in BibSonomy.

Lastly in Chapter 5, we proposed to discover trends from the social annotation temporal profiles. This is a novel task we identified to enhance object selection by incorporating non-trivial temporal features of the annotated objects. We demonstrated the utility of analyzing such non-trivial temporal features for scientific publications. We proposed a trend discovery process that enabled us to perform topic-specific and resource-specific trend analysis tasks. We proposed a trend estimation method that allowed us to parameterize and compare the emerging trends from the annotation temporal profiles. Although our study focused on the annotation profiles for scientific publications, the proposed trend discovery process and trend estimation methods can be applied directly to other types of temporal profiles, as noted in Section 5.2.1.

We experimented with two forms of social annotations for scientific publications, namely social tags and citing documents, to study the different trends in these two social communities. We evaluated our proposed trend discovery process and trend estimation method by performing a range of trend analysis tasks. These tasks include: (i) discovery of emerging topic trends; (ii) selection of important publications for given topics, as well as selection of emerging topics for given publications; (iii) identification of the most influential papers using trend-based metrics. The corpus-wise topic trends suggested reasonably significant topics rising from the research community in last decade. For two identified emerging topics, we examined the important publications being cited for these topics. For two identified publications, we examined the item-wise emerging topics citing these publications. One is a theory-oriented book, two groups of citing topics showed contrasting patterns of emergence. Another is an application case study paper, groups of topics showing similar emergence patterns are reasonably coherent. When comparing the two types of social annotations, we found that the temporal profiles formed by social tags are much sparser than those formed by citing documents for the same set of publications. Since trend analysis tasks consume tagging data, tags sparseness is again seen critical for these tasks.

In summary, this dissertation reported our studies on link prediction and object selection to facilitate navigation in social tagging systems. Our philosophy is that, there are three dimensions for navigating the social tagging space, namely the *semantic*, *social*, and *temporal* dimensions, and prediction tasks for link prediction and object selection can benefit from exploiting these dimensions.

6.2 Suggestions for Future Research

We identify the following possible future extensions for the tasks reported in this thesis.

Tag Prediction with Multiple Levels of Specificity

One characteristic of the proposed LDAtgg model is that topics are flat. In other words, the topics are assumed independent, and within each topic, the tags are associated with the topic independent with other tags. From the experimental results shown in Chapter 3, we found that, tags that are seen more often in the training set are more likely to be predicted by LDAtgg model. As a result, the model achieved good performance on documents with less exclusive tags, *i.e.* tags assigned to many documents, but poorer performance on documents with more exclusive tags. In general, tags of higher exclusiveness are more *specific* to the annotated resource. Hence, one possible improvement is to incorporate multiple levels of specificity.

To predict tags with multiple levels of specificity, two approaches are possible. The first approach is to exploit the semantic relations between tags within each topic [123]. To do that, additional sources of semantic information may be consulted from WordNet [25] or Wikipedia[1] [114] to improve the *bag-of-tags* representations in LDAtgg model. Second is to model topics with multiple levels of specificity, *e.g.* to learn a topic hierarchy using non-parametric models [8].

Tag Prediction for Non-textual Resources

LDAtgg model can also be extended to suit resource representations other than text. For instance, the image retrieval community has studied using social tags as image annotations for indexing and retrieval [69, 68, 21]. The ability to predict tags for untagged images, or multimedia content in general, would benefit such retrieval tasks. The key to extend LDAtgg model for multimedia content is to incorporate the suitable representations for the content features. In the setting of social tagging, such as Flickr and YouTube, interpreting the relevance of a user-created tag with respect to the visual content of a resource also imposes new challenges for such tasks.

[1] www.wikipedia.org

Tag Prediction using Trends

A promising direction to improve tag prediction is to consider the temporal aspect of social tagging data. We note that people annotate resources based on their current interest or current events, such as worldup2010 or iphone4. Hence, being able to capture the current trends in the social tagging community may lead to promising results in the prediction performance. The trend estimator we proposed in Chapter 5 may serve to bridge the two tasks.

Personalized Tag Recommendation using Topic Models

Adopting topic models for personalized tag recommendation may lead to promising results. For this direction, we foresee two challenges. The first challenge is to model *user-specific* topic multinomials and their relationships with the *community* topic multinomials. For the task of tag prediction studied in Chapter 3, we modeled the *community* topic multinomials despite which tags are used by which user. Such information may not be critical to tag prediction, but is critical to personalized tag recommendation. The second challenge is to learn topics for the target resource *on-demand*. Since the task of tag recommendation is, in general, more time-critical than the task of tag prediction, adopting topic models for personalized tag recommendation should be able to estimate the topic mixture for resources

Personalized Resource Search

One dual problem to personalized tag recommendation is personalized resource search in social tagging systems. The objective is to retrieve target resource(s) that the query user will annotate with the query tag. Formulated also as a probabilistic ranking problem, it is to estimate the probability $p\,(r|u_q, t_q)$. In this case, the query tag is a tag in the query user's personomy. For retrieving the relevant resources, we now translate the personomy tag to resource tags.

Following the same intuition on borrowing translations from neighbors, we can also devise a probabilistic framework as written in Eq. 6.1.

$$p\left(r|u_q,t_q\right) = \frac{\sum_u sim\left(u,u_q\right) \times p\left(r|u,t_q\right)}{\sum_u sim\left(u,u_q\right)} \qquad \text{(Eq. 6.1)}$$

We may adopt L1-norm and JS-divergence metrics to compute the similarity between users or explore other metrics [64]. To maximally reach candidate resources, we first translate personomy tags to resource tags (Eq. 6.3), and search for resources that have been assigned the translated tag (Eq. 6.2).

$$p\left(r|u,t_q\right) = \sum_{t_r \in \mathbb{T}} p\left(r|t_r\right) \times p\left(t_r|u,t_q\right) \qquad \text{(Eq. 6.2)}$$

$$p\left(t_r|u,t_q\right) = \sum_{r \in \mathbf{r}_u} p\left(t_q|r,u\right) \times p\left(t_r|r\right) \qquad \text{(Eq. 6.3)}$$

This problem, however, raises challenges in the evaluation in offline settings. To quantitatively evaluate the performance on resource search, ground truth should be given as the set of relevant resources for the query user when he/she meant to annotate with the query tag. In an offline setting, where evaluation is conducted on a static snapshot of the social tagging system, we cannot conclude irrelevance for resources not bookmarked by the query user. In other words, the resources not yet bookmarked by the query user are not judged for relevance. An online setting can be adopted to address this problem, where judgment can be collected from users on a real-time basis. However, conducting online judgment is time-consuming and sometimes involve monetary cost.

Suggestions for Trend Discovery

In Chapter 5, we use social annotations to profile scientific publications for trend discovery. Leveraging topic modeling, we derive topic series from the annotation profiles and perform trend estimation to find emerging topic trends. With the discovered trends, we are able to perform analysis tasks for understanding the scientific literature.

In the evaluation on the discovered trends, we face a number of challenges. Unlike TDT tasks in TREC, where test documents have manually labeled by human subjects, quantitative evaluation can be conducted. Direct evaluation on topic trends in scientific literature have been mostly qualitative [11, 44, 122]. In Chapter 5, we demonstrated a task to identify the most influential papers for evaluating the discovered emerging trend indirectly.

Given the proposed trend estimator, we are able to explore more interesting trend analysis tasks for understanding the scientific literature using social annotations. One task is to find (latent) connections between topics. Aside from using content-based similarity and direct citation links, one may use emerging trends to find *co-emerging* topics. Another interesting task is to discover *late-emerging topics* for individual publications. Such topics will be useful to understand the re-discovered value a publication has in new research specialties. To formally define and solve these tasks are future steps forward.

References

[1] Hend S. Al-Khalifa and Hugh C. Davis. Exploring the value of folksonomies for creating semantic metadata. *International Journal on Semantic Web and Information Systems*, 3(1):13–39, March 2007.

[2] James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. Topic detection and tracking pilot study final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218, 1998.

[3] Loulwah AlSumait, Daniel Barbará, and Carlotta Domeniconi. On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *Proceedings of the 8th IEEE International Conference on Data Mining*, pages 3–12. IEEE Computer Society, 2008.

[4] Morgan Ames and Mor Naaman. Why we tag: motivations for annotation in mobile and online media. In *Proceedings of the 25th Annual SIGCHI Conference on Human Factors in Computing Systems*, pages 971–980. ACM, 2007.

[5] Ching-man Au Yeung, Nicholas Gibbins, and Nigel Shadbolt. User-induced links in collaborative tagging systems. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management*, pages 787–796. ACM, 2009.

[6] Fabiano Muniz Belém, Eder Ferreira Martins, Jussara Marques Almeida, Marcos André Gonçalves, and Gisele Lobo Pappa. Exploiting co-occurrence and information quality metrics to recommend tags in Web 2.0 applications. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 1793–1796. ACM, 2010.

[7] Kerstin Bischoff, Claudiu S. Firan, Wolfgang Nejdl, and Raluca Paiu. Can all tags be used for search? In *Proceeding of the 17th ACM Conference on Information and Knowledge Management*, pages 193–202. ACM, 2008.

[8] David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems*. MIT Press, 2003.

[9] David M. Blei and Michael I. Jordan. Modeling annotated data. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 127–134. ACM, 2003.

[10] David M. Blei and John D. Lafferty. Correlated topic models. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 147–154. MIT Press, 2005.

[11] David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 113–120. ACM, 2006.

[12] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003.

[13] Toine Bogers and Antal van den Bosch. Recommending scientific articles using CiteULike. In *Proceedings of the 2nd ACM Conference on Recommender Systems*, pages 287–290. ACM, 2008.

[14] Levent Bolelli, Şeyda Ertekin, and C. Lee Giles. Topic and trend detection in text collections using latent dirichlet allocation. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, pages 776–780. Springer-Verlag, 2009.

[15] Levent Bolelli, Seyda Ertekin, Ding Zhou, and C. Lee Giles. Finding topic trends in digital libraries. In *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 69–72. ACM, 2009.

[16] F. Brugnara, D. Falavigna, and M. Omologo. Automatic segmentation and labeling of speech based on hidden Markov models. *Speech Communication*, 12(4):357–370, 1993.

[17] Adriana Budura, Sebastian Michel, Philippe Cudré-Mauroux, and Karl Aberer. To tag or not to tag-: Harvesting adjacent metadata in large-scale tagging systems. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 733–734. ACM, 2008.

[18] Markus Bundschus, Shipeng Yu, Volker Tresp, Achim Rettinger, Mathaeus Dejori, and Hans-Peter Kriegel. Hierarchical Bayesian models for collaborative tagging systems. In *Proceedings of the 9th IEEE International Conference on Data Mining*, pages 728–733. IEEE Computer Society, 2009.

[19] Chaomei Chen. CiteSpace II: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for Information Science and Technology*, 57:359–377, February 2006.

[20] Ed H. Chi and Todd Mytkowicz. Understanding the efficiency of social tagging systems using information theory. In *Proceedings of the 19th ACM Conference on Hypertext and Hypermedia*, pages 81–88. ACM, 2008.

[21] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Survey*, 40:1–60, May 2008.

[22] Ernesto Diaz-Aviles, Mihai Georgescu, Avaré Stewart, and Wolfgang Nejdl. LDA for on-the-fly auto tagging. In *Proceedings of the 4th ACM Conference on Recommender Systems*, pages 309–312. ACM, 2010.

[23] Pual Dourish and Matthew Chalmers. Running out of space: Models of information navigation. In Gilbert Cockton, Stephen W. Draper, and George R. S. Weir, editors, *Proceedings of the 11th Conference of the British Computer Society Human-Computer Interaction Specialist Group*. Cambridge University Press, August 1994.

[24] Umer Farooq, Thomas G. Kannampallil, Yang Song, Craig H. Ganoe, John M. Carroll, and Lee Giles. Evaluating tagging behavior in social bookmarking systems: Metrics and design heuristics. In *Proceedings of the 2007 International ACM Conference on Supporting Group Work*, pages 351–360. ACM, 2007.

[25] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[26] G. David Jr. Forney. The Viterbi algorithm: A personal history. In *Proceedings of Viterbi Conference*, pages 1–8, 2005.

[27] Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. Domain-specific keyphrase extraction. pages 668–673. Morgan Kaufmann Publishers Inc., 1999.

[28] Nikhil Garg and Ingmar Weber. Personalized, interactive tag recommendation for Flickr. In *Proceedings of the 2nd ACM Conference on Recommender Systems*, pages 67–74. ACM, 2008.

[29] Jonathan Gemmell, Maryam Ramezani, Thomas Schimoler, Laura Christiansen, and Mobasher Bamshad. The impact of ambiguity and redundancy on tag recommendation in folksonomies. In *Proceedings of the 3rd ACM conference on Recommender Systems*, pages 45–52. ACM, 2009.

[30] Mark Girolami and Ata Kabán. On an equivalence between PLSI and LDA. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 433–434. ACM, 2003.

[31] Sharon Givon and Victor Lavrenko. Large scale book annotation with social tags. In *Proceedings of the 3rd AAAI International Conference on Weblogs and Social Media*, pages 210–213. AAAI, 2009.

[32] André Gohr, Alexander Hinneburg, René Schult, and Myra Spiliopoulou. Topic evolution in a stream of documents. In *Proceedings of the 9th SIAM International Conference on Data Mining*, pages 859–870. SIAM, 2009.

[33] Andre Gohr, Myra Spiliopoulou, and Alexander Hinneburg. Visually summarizing the evolution of documents under a social tag. In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, 2010.

[34] Scott A. Golder and Bernardo A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, 2006.

[35] Saurabh Goorha and Lyle Ungar. Discovery of significant emerging trends. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 57–64. ACM, 2010.

[36] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Supplement 1):5228–5235, April 2004.

[37] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Supplement 1):5228–5235, 2004.

[38] Ziyu Guan, Jiajun Bu, Qiaozhu Mei, Chun Chen, and Can Wang. Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 540–547. ACM, 2009.

[39] Marieke Guy and Tonkin Emma. Folksonomies: Tidying up tags? *D-Lib Magazine*, 12(1), January 2006.

[40] David Hall, Daniel Jurafsky, and Christopher D. Manning. Studying the history of ideas using topic models. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 363–371. ACL, 2008.

[41] Harry Halpin, Valentin Robu, and Hana Shepherd. The complex dynamics of collaborative tagging. In *Proceedings of the 16th International Conference on World Wide Web*, pages 211–220. ACM, 2007.

[42] Conor Hayes and Paolo Avesani. Using tags and clustering to identify topic-relevant blogs. In *Proceedings of the 1st International AAAI Conference on Weblogs and Social Media*. AAAI Press, 2007.

[43] Dan He and D. Stott Parker. Topic dynamics: An alternative model of bursts in streams of topics. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 443–452. ACM, 2010.

[44] Qi He, Bi Chen, Jian Pei, Baojun Qiu, Prasenjit Mitra, and Lee Giles. Detecting topic evolution in scientific literature: How can citations help? In *Proceeding of the 18th ACM Conference on Information and Knowledge Management*, pages 957–966. ACM, 2009.

[45] Gregor Heinrich. Parameter estimation for text analysis. Technical report, fraunhofer igd, University of Leipzig, August 2009.

[46] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Can social bookmarking improve Web search? In *Proceedings of the 1st International Conference on Web Search and Data Mining*, pages 195–206. ACM, 2008.

[47] Paul Heymann, Daniel Ramage, and Hector Garcia-Molina. Social tag prediction. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 531–538. ACM, 2008.

[48] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57. ACM, 1999.

[49] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. BibSonomy: A social bookmark and publication sharing system. In Moonjeong Kang, editor, *Proceedings of the Conceptual Structures Tool*

Interoperability Workshop at the 14th International Conference on Conceptual Structures, pages 87–102. Aalborg University Press, 2006.

[50] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Trend detection in folksonomies. In *Proceedings of the 1st International Conference on Semantics And Digital Media Technology*, volume 4306, pages 56–70. Springer, 2006.

[51] Ming-Hung Hsu, Yu-Hui Chang, and Hsin-Hsi Chen. Temporal correlation between social tags and emerging long-term trend detection. In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*. AAAI, 2010.

[52] Meiqun Hu. A topic modeling approach to social tag prediction. *Bulletin of IEEE Technical Committee on Digital Libraries*, 6, Fall 2010.

[53] David Hull. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 329–338. ACM, 1993.

[54] Sebastian Jaimungal and Eddie K. H. Ng. Consistent functional pca for financial time-series. In *Proceedings of the 4th IASTED International Conference on Financial Engineering and Applications*, pages 103–108. ACTA Press, 2007.

[55] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20:422–446, October 2002.

[56] Robert Jäschke, Folke Eisterlehner, Andreas Hotho, and Gerd Stumme. Testing and evaluating tag recommenders in a live system. In *Proceedings of the 3rd ACM Conference on Recommender Systems*, pages 369–372. ACM, 2009.

[57] Robert Jäschke, Leandro Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in folksonomies. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 506–514. Springer-Verlag, 2007.

[58] Said Kashoob, James Caverlee, and Ying Ding. A categorical model for discovering latent structure in social annotations. In *Proceedings of the 3rd International AAAI Conference on Weblogs and Social Media*. AAAI Press, 2009.

[59] Jon Kleinberg. Bursty and hierarchical structure in streams. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 91–101. ACM, 2002.

[60] Christian Körner, Dominik Benz, Andreas Hotho, Markus Strohmaier, and Stum Gerd. Stop thinking, start tagging: Tag semantics emerge from collaborative verbosity. In *Proceedings of the 19th International Conference on World Wide Web*, pages 521–530. ACM, 2010.

[61] Ralf Krestal, Peter Fankhauser, and Wolfgang Nejdl. Latent Dirichlet allocation for tag recommendation. In *Proceedings of the 3rd ACM Conference on Recommender Systems*, pages 61–68. ACM, 2009.

[62] Ralf Krestel and Peter Fankhauser. Tag recommendation using probabilistic topic models. In *ECML PKDD Discovery Challenge 2009*, volume 497, pages 131–141. CEUR Workshop Proceedings, 2009.

[63] Ralf Krestel and Peter Fankhauser. Language models and topic models for personalizing tag recommendation. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, pages 82–89. IEEE Computer Society, 2010.

[64] Lillian Lee. *Similarity-Based Approaches to Natural Language Processing.* Doctoral thesis, Harvard University, Cambridge, MA, USA, 1997. Chapter Four.

[65] Lillian Lee. Measures of distributional similarity. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 25–32. ACL, 1999.

[66] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 497–506. ACM, 2009.

[67] Xin Li, Lei Guo, and Yihong Eric Zhao. Tag-based social interest discovery. In *Proceeding of the 17th International Conference on World Wide Web*, pages 675–684. ACM, 2008.

[68] Xirong Li, Cees G. M. Snoek, and Marcel Worring. Unsupervised multi-feature tag relevance learning for social image retrieval. In *Proceedings of the 9th ACM International Conference on Image and Video Retrieval*, pages 10–17. ACM, 2010.

[69] Xirong Li, Cees G.M. Snoek, and Marcel Worring. Learning tag relevance by neighbor voting for social image retrieval. In *Proceeding of the 1st ACM International Conference on Multimedia Information Retrieval*, pages 180–187. ACM, 2008.

[70] Xia Lin, Joan E. Beaudoin, Yen Bul, and Kushal Desal. Exploring characteristics of social classification. In Jonathan Furner and Joseph T. Tennis, editors, *Proceedings 17th Workshop of the American Society for Information Science and Technology Special Interest Group in Classification Research*, volume 17, 2006.

[71] Caimei Lu, Xiaohua Hu, Xin Chen, Jung-Ran Park, TingTing He, and Zhoujun Li. The topic-perspective model for social tagging systems. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 683–692. ACM, 2010.

[72] Yu-Ta Lu, Shoou-I Yu, Tsung-Chieh Chang, and Jane Yung-jen Hsu. A content-based method to enhance tag recommendation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 2064–2069. Morgan Kaufmann Publishers Inc., 2009.

[73] George Macgregor and Emma McCulloch. Collaborative tagging as a knowledge organisation and resource discovery tool. *Library Review*, 55(5):291–300, 2006.

[74] Ketan K. Mane and Katy Börner. Mapping topics and topic bursts in pnas. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Supplement 1):5287–5290, 2004.

[75] Gideon S. Mann, David Mimno, and Andrew McCallum. Bibliometric impact measures leveraging topic analysis. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 65–74. ACM, 2006.

[76] Balby Leandro Marinho, Christine Preisach, and Lars Schmidt-Thieme. Relational classification for personalized tag recommendation. In *ECML PKDD Discovery Challenge 2009*, volume 497, pages 7–15. CEUR Workshop Proceedings, 2009.

[77] Leandro Marinho, B and Lars Schmidt-Thieme. *Collaborative Tag Recommendations*, chapter 63, pages 533–540. Springer Berlin Heidelberg, 2008.

[78] Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. HT06, tagging paper, taxonomy, Flickr, academic article, to read. In *Proceedings*

of the 17th Conference on Hypertext and Hypermedia, pages 31–40. ACM, 2006.

[79] A. Fani Marvasti and D. B. Skillicorn. Structures in collaborative tagging: An empirical analysis. In *Proceedings of the 33rd Australasian Conference on Computer Science*, volume 102, pages 109–116. Australian Computer Society, Inc., 2010.

[80] Adam Mathes. Folksonomies - cooperative classification and communication through shared metadata. *Journal of Computer Mediated Communication*, December 2004.

[81] Miller McPherson, Lynn Smith-Lovin, and James M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444, 2001.

[82] Olena Medelyan, Eibe Frank, and Ian H. Witten. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, volume 3, pages 1318–1327. ACL, 2009.

[83] Qiaozhu Mei, Chao Liu, Hang Su, and ChengXiang Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proceedings of the 15th International Conference on World Wide Web*, pages 533–542. ACM, 2006.

[84] Guilherme Vale Menezes, Jussara M. Almeida, Fabiano Belém, Marcos André Gonçalves, Anísio Lacerda, Edleno Silva De Moura, Gisele L. Pappa, Adriano Veloso, and Nivio Ziviani. Demand-driven tag recommendation. In *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, pages 402–417. Springer-Verlag, 2010.

[85] Fabian Mörchen, Mathäus Dejori, Dmitriy Fradkin, Julien Etienne, Bernd Wachmann, and Markus Bundschus. Anticipating annotations and emerging trends in biomedical literature. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 954–962. ACM, 2008.

[86] Satoshi Morinaga and Kenji Yamanishi. Tracking dynamics of topic trends using a finite mixture model. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 811–816. ACM, 2004.

[87] Hendri Murfi and Klaus Obermayer. A two-level learning hierarchy of concept based keyword extraction for tag recommendation. In *ECML PKDD Discovery Challenge 2009*, volume 497, pages 201–214. CEUR Workshop Proceedings, 2009.

[88] David Newman, Chaitanya Chemudugunta, and Padhraic Smyth. Statistical entity-topic models. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 680–686. ACM, 2008.

[89] Oded Nov, Mor Naaman, and Chen Ye. What drives content tagging: the case of photos on Flickr. In *Proceeding of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems*, pages 1097–1100. ACM, 2008.

[90] The ACM Special Interest Group on Software Engineering. ICSE's most influential paper award. Online. Retrieved on April 2011.

[91] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.

[92] Denis Parra and Peter Brusilovsky. Collaborative filtering for social tagging systems: an experiment with CiteULike. In *Proceedings of the 3rd ACM Conference on Recommender Systems*, pages 237–240. ACM, 2009.

[93] Emilee Rader and Rick Wash. Influences on tag choices in del.icio.us. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, pages 239–248. ACM, 2008.

[94] Adam Rae, Börkur Sigurbjörnsson, and Roelof van Zwol. Improving tag recommendation using social networks. In *Adaptivity, Personalization and Fusion of Heterogeneous Information*, pages 92–99. Le Centre de Hautes Etudes Internationales d'Informatique Documentaire, 2010.

[95] Steffen Rendle and Schmidt-Thie Lars. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, pages 81–90. ACM, 2010.

[96] Valentin Robu, Harry Halpin, and Hana Shepherd. Emergence of consensus and shared vocabularies in collaborative tagging systems. *ACM Transactions on the Web*, 3:14:1–14:34, September 2009.

[97] Luis Sarmento, Sérgio Nunes, Jorge Teixeira, and Eugénio Oliveira. Propagating fine-grained topic labels in news snippets. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*, pages 515–518. IEEE Computer Society, 2009.

[98] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. CROC: A new evaluation criterion for recommender systems. *Journal of Electronic Commerce Research*, 5:51–74, January 2005.

[99] Shilad Sen, Shyong K. Lam, Al Mamunur Rashid, Dan Cosley, Dan Frankowski, Jeremy Osterhouse, F. Maxwell Harper, and John Riedl.

Tagging, communities, vocabulary, evolution. In *Proceedings of the 20th Anniversary Conference on Computer Supported Cooperative Work*, pages 181–190. ACM.

[100] Shilad Sen, Jesse Vig, and John Riedl. Tagommenders: Connecting users to items through tags. In *Proceedings of the 18th International Conference on World Wide Web*, pages 671–680. ACM, 2009.

[101] Clay Shirky. Folksonomy, August 2004.

[102] Xiance Si and Maosong Sun. Tag-LDA for scalable real-time tag recommendation. *Journal of Computational Information Systems*, 6(1):23–31, 2009.

[103] Börkur Sigurbjörnsson and Roelof van Zwol. Flickr tag recommendation based on collective knowledge. In *Proceeding of the 17th International Conference on World Wide Web*, pages 327–336. ACM, 2008.

[104] T Smith. Cataloging and you: Measuring the efficacy of a folksonomy for subject analysis. *Digital Library of Information Science and Technology*, October 2007.

[105] Yang Song, Ziming Zhuang, Huajing Li, Qiankun Zhao, Jia Li, Wang-Chien Lee, and C. Lee Giles. Real-time automatic tag recommendation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 515–522. ACM, 2008.

[106] Myra Spiliopoulou, Irene Ntoutsi, Yannis Theodoridis, and Rene Schult. MONIC: Modeling and monitoring cluster transitions. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 706–711. ACM, 2006.

[107] Shankara B. Subramanya and Huan Liu. SocialTagger - collaborative tagging for blogs in the long tail. In *Proceedings of the 2008 ACM Workshop on Search in Social Media*, pages 19–26. ACM, 2008.

[108] Ke Sun, Xiaolong Wang, Chengjie Sun, and Lei Lin. A language model approach for tag recommendation. *Journal of Expert Systems with Applications*, 38:1575–1582, March 2011.

[109] Martin Svensson. *Defining, Designing and Evaluating Social Navigation.* Doctoral thesis, University of Stockholm, 2003. Chapter Two.

[110] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *Proceedings of the 2nd ACM Conference on Recommender Systems*, pages 43–50. ACM, 2008.

[111] Jennifer Trant. Studying social tagging and folksonomy: A review and framework. *Journal of Digital Information*, 10(1):1–44, 2009.

[112] Karen H. L. Tso-Sutter, Leandro Balby Marinho, and Schmidt-Thie Lars. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proceedings of the 23rd ACM Symposium on Applied Computing*, pages 1995–1999. ACM, 2008.

[113] Thomas Vander Wal. Folksonomy coinage and definition, Februray 2007.

[114] Max Völkel, Markus Krötzsch, Denny Vrandecic, Heiko Haller, and Rudi Studer. Semantic Wikipedia. In *Proceedings of the 15th International Conference on World Wide Web*, pages 585–594. ACM, 2006.

[115] Jian Wang and Brian D. Davison. Explorations in tag suggestion and query expansion. In *Proceedings of the 2008 ACM Workshop on Search in Social Media*, pages 43–50, New York, NY, USA, 2008.

[116] Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 697–702. IEEE Computer Society, 2007.

[117] Robert Wetzker, Till Plumbaum, Alexander Korth, Christian Bauckhage, Tansu Alpcan, and Florian Metze. Detecting trends in social bookmarking systems using a probabilistic generative model and smoothing. In *Proceedings of the 19th International Conference on Pattern Recognition*, pages 1–4. IEEE Computer Society, 2008.

[118] Robert Wetzker, Alan Said, and Carsten Zimmermann. Understanding the user: Personomy translation for tag recommendation. In *ECML PKDD Discovery Challenge 2009*, volume 497, pages 275–284. CEUR Workshop Proceedings, 2009.

[119] Robert Wetzker, Carsten Zimmermann, and Christian Bauckhage. Analyzing social bookmarking systems: A del.icio.us cookbook. In *Proceedings of the European Conference on Artificial Intelligence*, pages 26–30. IOS Press, 2008.

[120] Robert Wetzker, Carsten Zimmermann, Christian Bauckhage, and Sahin Albayrak. I tag, you tag: Translating tags for advanced user models. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, pages 71–80. ACM, 2010.

[121] Zhijun Yin, Rui Li, Qiaozhu Mei, and Jiawei Han. Exploring social gagging graph for Web object classification. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 957–966. ACM, 2009.

[122] Ding Zhou, Xiang Ji, Hongyuan Zha, and C. Lee Giles. Topic evolution and social interactions: How authors effect research. In *Proceedings of*

the 15th ACM International Conference on Information and Knowledge Management, pages 248–257. ACM, 2006.

[123] Tom Chao Zhou and Irwin King. Automobile, car and BMW: Horizontal and hierarchical approach in social tagging systems. In *Proceedings of the 2nd ACM Workshop on Social Web Search and Mining*, pages 25–32. ACM, 2009.

[124] Yunyue Zhu and Dennis Shasha. Efficient elastic burst detection in data streams. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 336–345. ACM, 2003.

Appendix A

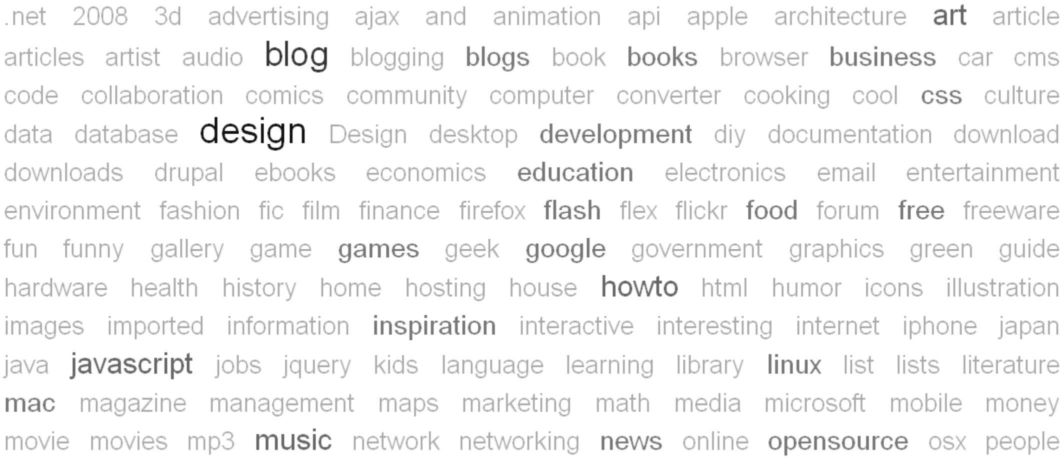
Navigational Views on Social Tagging Systems

There are many popular social tagging sites in existence. For instances, Delicious is the largest social tagging site to date for annotating web URLs; CiteULike[1] and BibSonomy[2] are designed for annotating scientific references for the research community; LibraryThing[3] and Goodreads[4] allow users to create their own library catalogs for books; Last.fm[5] supports users to tag artists, albums and songs. Social tagging systems can be a web site by its own, such as the above mentioned, where the content of the web resources are not hosted on the tagging site. Social tagging can also be embedded in other content-publishing sites, such as Flickr[6] and YouTube[7], where tags serve as metadata to organize and identify the published content. Besides the basic function of tagging, many existing social tagging systems also provide aggregated views of tagging data, such as the following, to help users navigate the social tagging space.

Tag cloud presents a visualization to feature tags used for a resource

[1] www.citeulike.org
[2] www.bibsonomy.org
[3] www.librarything.com
[4] www.goodreads.com
[5] www.last.fm
[6] www.flickr.com
[7] www.youtube.com

collection. The more frequently used tags are shown in larger font in the cloud. Figure A.1 shows a tag cloud from Delicious. By selecting a tag in the cloud, one can examine all resources annotated with the selected tag.



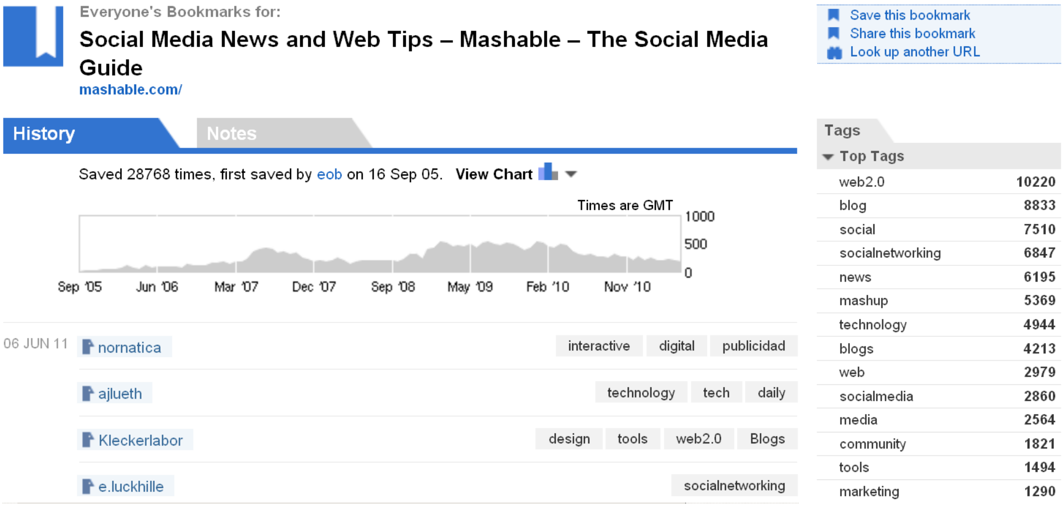
.net 2008 3d advertising ajax and animation api apple architecture art article
articles artist audio blog blogging blogs book books browser business car cms
code collaboration comics community computer converter cooking cool css culture
data database design Design desktop development diy documentation download
downloads drupal ebooks economics education electronics email entertainment
environment fashion fic film finance firefox flash flex flickr food forum free freeware
fun funny gallery game games geek google government graphics green guide
hardware health history home hosting house howto html humor icons illustration
images imported information inspiration interactive interesting internet iphone japan
java javascript jobs jquery kids language learning library linux list lists literature
mac magazine management maps marketing math media microsoft mobile money
movie movies mp3 music network networking news online opensource osx people

Figure A.1: An Example of Tag Cloud

Tag view presents a combined view of everyone's bookmarks for a given tag. Figure 1.1 in Chapter 1 shows an example tag view for the tag `socialmedia` on Delicious. It allows the users to explore resources that other users have bookmarked with the given tag. Furthermore, resources are ordered by the number of users using the given tag for annotating them.

Resource view presents a combined view of everyone's bookmarks for a given resource, such as the one shown in Figure A.2. This resource view has two parts. On the left hand side, it shows a link to the user by whom the resource was first bookmarked, plots the volume of bookmarks received over time, and lists the most recent bookmarks (cum users and their tag assignments) the given resource has received. On the right hand side, it summarizes the most frequent tags assigned to the resource. Provided that there has been adequate bookmarks to the given resource, these top frequent tags are regarded as the primary semantics of the resource, representing the consensus from the social tagging community [34, 99].

User view presents the collection of resources bookmarked by the given user and the collection of tags used by the user for annotating resources, such



Everyone's Bookmarks for:

Social Media News and Web Tips – Mashable – The Social Media Guide

mashable.com/

Save this bookmark
Share this bookmark
Look up another URL

History | Notes

Saved 28768 times, first saved by eob on 16 Sep 05. View Chart

06 JUN 11	nornatica	interactive digital publicidad
	ajlueth	technology tech daily
	Kleckerlabor	design tools web2.0 Blogs
	e.luckhille	socialnetworking

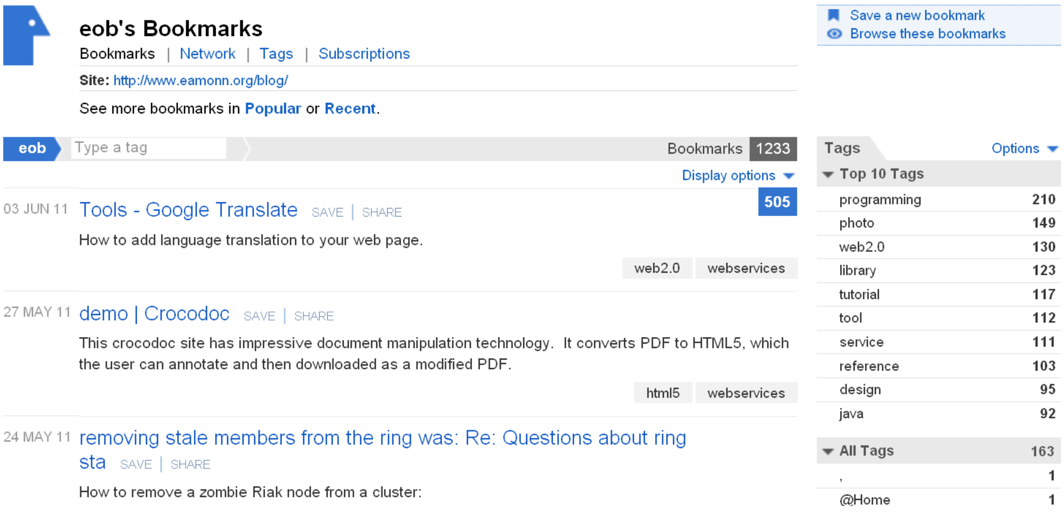
Tags

Top Tags

web2.0	10220
blog	8833
social	7510
socialnetworking	6847
news	6195
mashup	5369
technology	4944
blogs	4213
web	2979
socialmedia	2860
media	2564
community	1821
tools	1494
marketing	1290

Figure A.2: An Example of Resource View

as the example shown in Figure A.3. On Delicious, everyone's bookmarks are visible to everyone else, unless a user explicitly specifies a bookmark to be private. Finding resources through viewing others' bookmarks is analogous to *watching and following* others or taking their advices. This is, as noted by Dourish and Chalmers in [23] and Svensson in [109], a form of social navigation.



eob's Bookmarks

Bookmarks | Network | Tags | Subscriptions

Site: http://www.eamonn.org/blog/

See more bookmarks in Popular or Recent.

Save a new bookmark
Browse these bookmarks

eob | Type a tag | Bookmarks 1233

Display options

03 JUN 11 Tools - Google Translate SAVE | SHARE 505

How to add language translation to your web page.

web2.0 webservices

27 MAY 11 demo | Crocodoc SAVE | SHARE

This crocodoc site has impressive document manipulation technology. It converts PDF to HTML5, which the user can annotate and then downloaded as a modified PDF.

html5 webservices

24 MAY 11 removing stale members from the ring was: Re: Questions about ring sta SAVE | SHARE

How to remove a zombie Riak node from a cluster:

Tags Options

Top 10 Tags

programming	210
photo	149
web2.0	130
library	123
tutorial	117
tool	112
service	111
reference	103
design	95
java	92

All Tags 163

,	1
@Home	1

Figure A.3: An Example of User View

Tag recommendation is provided to help a user choose the appropriate tags when bookmarking a resource. Figure 4.1 in Chapter 4 shows an example of tag recommendations on Delicious. Often, the recommended tags are those most frequently assigned by other users annotating the given resource (if there has been any). This, again, demonstrates social navigation, where the recommended tags are collected from other users in the social tagging community.

Resource recommendation recommends new resources to users that aims to meet their interest in consumption. For example, CiteULike recommends scientific articles not yet annotated by the user, and Last.fm recommends music playlist to users. Such recommendations are often based on the annotation profiles of the users and the tags associated with the resources [13, 92, 100].

In summary, these navigational views are built upon the assignment relationships between information objects in the social tagging space to meet diverse user needs. They unveil the semantic, social, or temporal attributes of the current object(s) by aggregating and ordering the links from the current object(s) to the most emergent others. They enable users to navigate the social tagging space from the presented views.

Appendix B

Conditional Probabilities in LDAtgg Model

In what follows, we lay out the derivations of Eq. 3.2 and Eq. 3.3, which computes the joint a posteriori probabilities of topics for word tokens and tag tokens respectively in Algorithm 1. These derivations are largely guided by the technical report by Heinrich [45]. We follow the same set of notations shown in Table 3.3.

B.1 Sampling Topics for Word Tokens

$$\begin{aligned} & p\left(z_{d,i}|\vec{z}_{\neg\{d,i\}}, \vec{w}, \vec{y}, \vec{t}\right) \\ = & \frac{p\left(z_{d,i}, \vec{z}_{\neg\{d,i\}}, \vec{w}, \vec{y}, \vec{t}\right)}{p\left(\vec{z}_{\neg\{d,i\}}, \vec{w}, \vec{y}, \vec{t}\right)} \\ = & \frac{p\left(\vec{z}\right) p\left(\vec{w}|\vec{z}\right) p\left(\vec{y}|\vec{z}\right) p\left(\vec{t}|\vec{y}\right)}{p\left(\vec{z}_{\neg\{d,i\}}\right) p\left(\vec{w}|\vec{z}_{\neg\{d,i\}}\right) p\left(\vec{y}|\vec{z}_{\neg\{d,i\}}\right) p\left(\vec{t}|\vec{y}\right)} \\ = & \frac{p\left(\vec{z}\right)}{p\left(\vec{z}_{\neg\{d,i\}}\right)} \times \frac{p\left(\vec{w}|\vec{z}\right)}{p\left(\vec{w}|\vec{z}_{\neg\{d,i\}}\right)} \times \frac{p\left(\vec{y}|\vec{z}\right)}{p\left(\vec{y}|\vec{z}_{\neg\{d,i\}}\right)} \qquad \text{(Eq. B.1)} \end{aligned}$$

Let function $\Delta()$ denote the Dirichlet integral [45]. We express the three components in Eq. B.1 as follows:

$$\begin{aligned}\frac{p(\vec{z})}{p\left(\vec{z}_{\neg\{d,i\}}\right)} &\propto \frac{\Delta(\vec{n}_d+\vec{\alpha})}{\Delta\left(\vec{n}_{d,\neg\{d,i\}}+\vec{\alpha}\right)} \\ &\propto \frac{n_{d,\neg i}^{z_{d,i}}+\alpha}{\sum_{k=1}^{K}\left(n_d^k+\alpha\right)-1} \qquad \text{(Eq. B.2)}\end{aligned}$$

$$\begin{aligned}\frac{p(\vec{w}|\vec{z})}{p\left(\vec{w}|\vec{z}_{\neg\{d,i\}}\right)} &= \frac{p(\vec{w}|\vec{z})}{p\left(\vec{w}_{\neg\{d,i\}}|\vec{z}_{\neg\{d,i\}}\right)p\left(w_{d,i}|\vec{z}_{\neg\{d,i\}}\right)} \\ &\propto \frac{\Delta\left(\vec{n}_k+\vec{\beta}\right)}{\Delta\left(\vec{n}_{k,\neg\{d,i\}}+\vec{\beta}\right)} \\ &\propto \frac{n_{k,\neg\{d,i\}}^{w_{d,i}}+\beta}{\sum_{w=1}^{W}\left(n_{k,\neg\{d,i\}}^{w}+\beta\right)} \qquad \text{(Eq. B.3)}\end{aligned}$$

$$\begin{aligned}\frac{p(\vec{y}|\vec{z})}{p\left(\vec{y}|\vec{z}_{\neg\{d,i\}}\right)} = \frac{p(\vec{y}_d|\vec{z}_d)}{p(\vec{y}_d|\vec{z}_{d,\neg i})} &\propto p(\vec{y}_d|\vec{z}_d) \\ &= \prod_{j=1}^{J_d} p(y_{d,j}|\vec{z}_d) \\ &= \prod_{j=1}^{J_d} \frac{n_d^{y_{d,j}}}{I_d} \qquad \text{(Eq. B.4)}\end{aligned}$$

Let m_d^k denote the number of tag tokens in document d that is assigned to topic k. Thus, Eq. B.4 can be re-written as:

$$\frac{p(\vec{y}|\vec{z})}{p\left(\vec{y}|\vec{z}_{\neg\{d,i\}}\right)} = \prod_{j=1}^{J_d}\frac{n_d^{y_{d,j}}}{I_d} = \prod_{k=1}^{K}\left(\frac{n_d^k}{I_d}\right)^{m_d^k} \qquad \text{(Eq. B.5)}$$

During sampling, instead of having n_d^k, we have $n_{d,\neg i}^k$ for each particular topic k. Hence, when computing $p\left(z_{d,i}=k|\vec{z}_{\neg\{d,i\}},\vec{w},\vec{y},\vec{t}\right)$ for a given k, we should assign $n_d^k := 1+n_{d,\neg i}^k$, and keep the counts for all other topics unchanged. Let $\bar{\mathcal{P}}$ denote the product $\prod_{k=1}^{K}\left(\frac{n_{d,\neg i}^k}{I_d}\right)^{m_d^k}$, then at $p\left(z_{d,i}=k|\vec{z}_{\neg\{d,i\}},\vec{w},\vec{y},\vec{t}\right)$ we have,

$$\prod_{k=1}^{K}\left(\frac{n_d^k}{I_d}\right)^{m_d^k} = \bar{\mathcal{P}}\times\left(\frac{1+n_{d,\neg i}^k}{n_{d,\neg i}^k}\right)^{m_d^k} \qquad \text{(Eq. B.6)}$$

The term $\bar{\mathcal{P}}$ in Equation Eq. B.6 can omitted in the computation, since it is constant for all k. We substitute Eq. B.2, Eq. B.3 and Eq. B.6 into Eq. B.1, then we have Eq. 3.2 shown in Algorithm 1.

B.2 Sampling Topics for Tag Tokens

$$\begin{aligned}
& p\left(y_{d,j}|\vec{y}_{\neg\{d,j\}},\vec{t},\vec{z},\vec{w}\right) \\
= & \frac{p\left(y_{d,j},\vec{y}_{\neg\{d,j\}},\vec{t},\vec{z},\vec{w}\right)}{p\left(\vec{y}_{\neg\{d,j\}},\vec{t},\vec{z},\vec{w}\right)} \\
= & \frac{p\left(\vec{z}\right)p\left(\vec{w}|\vec{z}\right)p\left(\vec{y}|\vec{z}\right)p\left(\vec{t}|\vec{y}\right)}{p\left(\vec{z}\right)p\left(\vec{w}|\vec{z}\right)p\left(\vec{y}_{\neg\{d,j\}}|\vec{z}\right)p\left(\vec{t}|\vec{y}_{\neg\{d,j\}}\right)} \\
= & \frac{p\left(\vec{y}|\vec{z}\right)}{p\left(\vec{y}_{\neg\{d,j\}}|\vec{z}\right)}\times\frac{p\left(\vec{t}|\vec{y}\right)}{p\left(\vec{t}|\vec{y}_{\neg\{d,j\}}\right)} \qquad \text{(Eq. B.7)}
\end{aligned}$$

$$\begin{aligned}
\frac{p\left(\vec{y}|\vec{z}\right)}{p\left(\vec{y}_{\neg\{d,j\}}|\vec{z}\right)}=\frac{p\left(\vec{y}_{\neg\{d,j\}}|\vec{z}\right)p\left(y_{d,j}|\vec{z}\right)}{p\left(\vec{y}_{\neg\{d,j\}}|\vec{z}\right)} & = p\left(y_{d,j}|\vec{z}_d\right) \\
& = \frac{n_d^{y_{d,j}}}{I_d} \qquad \text{(Eq. B.8)}
\end{aligned}$$

$$\begin{aligned}
\frac{p\left(\vec{t}|\vec{y}\right)}{p\left(\vec{t}|\vec{y}_{\neg\{d,j\}}\right)} & = \frac{p\left(\vec{t}|\vec{y}\right)}{p\left(\vec{t}_{\neg\{d,j\}}|\vec{y}_{\neg\{d,j\}}\right)p\left(t_{d,j}|\vec{y}_{\neg\{d,j\}}\right)} \\
& = \frac{p\left(\vec{t}_d|\vec{y}_d\right)}{p\left(\vec{t}_{d,\neg j}|\vec{y}_{d,\neg j}\right)} \\
& \propto \frac{\Delta\left(\vec{m}_y+\vec{\gamma}\right)}{\Delta\left(\vec{m}_{y,\neg\{d,j\}}+\vec{\gamma}\right)} \\
& \propto \frac{m_{k,\neg\{d,j\}}^{t_{d,j}}+\gamma}{\sum_{t=1}^{T}\left(m_{k,\neg\{d,j\}}^{t}+\gamma\right)} \qquad \text{(Eq. B.9)}
\end{aligned}$$

We substitute Eq. B.8 and Eq. B.9 into Eq. B.7, then we have Eq. 3.3 shown in Algorithm 1.