6-2016

User Behavior Mining in Microblogging

Tuan Anh HOANG
Singapore Management University, tahoang.2011@phdis.smu.edu.sg

Citation
HOANG, Tuan Anh. User Behavior Mining in Microblogging. (2016). 1-212.
Available at: https://ink.library.smu.edu.sg/etd_coll/130

USER BEHAVIOR MINING IN MICROBLOGGING

HOANG TUAN ANH

SINGAPORE MANAGEMENT UNIVERSITY
2015

User Behavior Mining in Microblogging

by

Hoang Tuan Anh

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

Dissertation Committee:

Ee-Peng Lim (Supervisor/Chair)
Professor of Information Systems
Singapore Management University

Jing Jiang
Assistant Professor of Information Systems
Singapore Management University

Baihua Zheng
Associate Professor of Information Systems
Singapore Management University

Wee Sun Lee
Professor
National University of Singapore

Singapore Management University
2015

User Behavior Mining in Microblogging

by

Hoang Tuan Anh

Abstract

This dissertation addresses the modeling of factors concerning microblogging users' content and behavior. We focus on two sets of factors. The first set includes behavioral factors of users and content items driving content propagation in microblogging. The second set consists of latent topics and communities of users as the users are engaged in content generation and behavior adoptions. These two sets of factors are extremely important in many applications, e.g., network monitoring and recommender systems.

In the first part of this dissertation, we identify user virality, user susceptibility, and content virality as three behavioral factors that affect users' behaviors in content propagation. User virality refers to the ability of a user in getting her content propagated by many other users, while user susceptibility refers to the tendency of a user to propagate other users' content. Content virality refers to the tendency of a content item to attract propagation by users. Instead of modeling these factors independently as done in previous research, we propose to jointly model all these factors considering their inter-relationships. We develop static, temporal, and incremental models for measuring the factors based on propagation data. We also develop a static model for modeling the factors specific to topics.

In the second part of this dissertation, we develop topic models for learning users' topical interest and communities from both their content and behavior. We first propose a model to derive community affiliations of users using topics and sentiments expressed in their content as well as their behavior. We then extend the model to learn both users' personal interest and that of their com-

munities, distinguishing the two types of interests. Our model also learns the bias of users toward their communities when generating content and adopting behavior.

Contents

List of Figures

List of Tables

Acknowledgements

I am grateful to my advisor, Professor Ee-Peng Lim, for his continuous support, guidance, and encouragement, and for providing me with the best working environment throughout my PhD study.

I owe my deep gratitude to Professor William W. Cohen for his supervision when I visited Carnegie Mellon University in 2012. Lessons I learnt from him play an important role in my works that form the second part of this dissertation.

I appreciate useful comments and feedbacks from members of my dissertation committee: Professor Ee-Peng Lim, Assistant Professor Jing Jiang, Associate Professor Baihua Zheng, and Professor Wee Sun Lee.

I were fortunate to have supports from my fellow researchers and friends in Living Analytics Research Centre (LARC): Palakorn Achananuparp, Freddy Chong Tat Chua, Dai Bing Tian, Philips K. Prasetyo, Agus T. Kwee, Ibrahim Nelman Lubis, Arinto Murdopo, Luu Minh Duc, and Gong Wei.

I am indebted to administrative staffs in School of Information Systems: Ong Chew Hong and Seow Pei Huan; and the staffs in LARC: Chua Kian Peng, Cheong Su Fen, Angela Kwek Renfeng, Sheryl Chen, Phoebe Yeo, Alenzia Wong Poh Luan, Fong Soon Keat, Desmond Yap, Janice Tan, and Nancy Beatty.

I would like to thank Singapore Management University for the scholarships and financial supports for my study. My research work is funded by The Singapore National Research Foundation under its International Research Centre @

Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority (MDA).

Dedicated to my family

Chapter 1

Introduction

1.1 Motivation

Microblogging sites such as Twitter[1] and Weibo[2] have become extremely popular due to the ease of posting short messages (called tweets) using both desktop and mobile devices. By the end of 2014, Twitter and Weibo have more than 300 millions[3] and 170 millions[4] monthly active users respectively. Users on microblogging sites interact with one another for various purposes, including information sharing [123], product broadcasting [101], political campaigning [69, 80, 262], and social mobilization [217, 211], etc..

Other than posting tweets, microblogging users may also adopt different types of behaviors. Table 1.1 shows the most common types of user behaviors in microblogging which also make microblogging unique compared with other social media. The behaviors are:

- **Relationship behavior**: These are the behaviors of users with regards to their relationships with other users. Examples of these behaviors are users *follow* other users, or *unfollow* some other users, etc..

- **Communication behavior**: These are behaviors of users when they

[1] https://twitter.com/

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

[3] https://about.twitter.com/company

[4] https://www.techinasia.com/weibo-2014-176-million-monthly-active-users/

Table 1.1: Common types of user behaviors in microblogging

Behavior type	Examples
Relationship	follow, unfollow other users
Communication	mention, reply other users
Propagation	retweet other users' tweets, share URL links
Linguistic	mention terms and hashtags

communicate with other users, including mentioning other users so as to direct tweets to the mentioned users, or replying other users' tweets, etc..

- **Propagation behavior**: These are behaviors of users as they adopt content items introduced to them by friends. Examples of this behavior include re-sharing a URL link from other users, reusing a hashtag that friend(s) have used before, and *retweeting* (forwarding) tweets posted by friend(s), etc..

- **Linguistic behavior**: This refers to behaviors of users in the choice of linguistic elements. For example, users may mention some terms in their biographies, or insert hashtags in their tweets, etc..

Due to the richness in content and user behavior data as well as many users sharing these data publicly, microblogging has also become a valuable data source to study user behavior and preference. In this dissertation, we focus on two research tasks. The first is to model the users' content propagation behavior, which is probably the most dominant behavior in microblogging. The second is to model user communities which have different behavior interests. We refer the first task as **modeling user behavior in content propagation**, and the second task as **modeling community behavior** respectively.

These two research tasks are extremely important in many applications. For example, content propagation is a key mechanism that supports advertisement and marketing [101], event detection [197, 217, 56, 55], rumors detection [153, 99, 181], and information credibility evaluation [32], etc.. Community behaviors are useful in election result prediction [213], user profiling [168, 29], and personalized recommendation [53, 77, 177], etc.. Both the tasks have been studied in a number of research projects. In our literature survey, we however found that the prior works suffer from two following major shortcomings.

Lack of latent factors in behavior modeling. Empirical works have shown that there are three behavioral factors that drive content propagation in microblogging [204, 33, 228, 188, 149, 18, 25, 259]. They are *user virality*, (b) *user susceptibility*, and (c) *content virality*. User virality refers to the ability of a user in getting propagation for her content items, while user susceptibility refers to the tendency of a user to adopt a propagation behavior on others' content. Content virality refers to the infectiousness of a content item in attracting propagation by users. Past studies have also shown that there are inter-relationships among these factors [228, 105, 47, 10, 203, 83]. Existing methods for measuring these factors however do not consider their inter-relationships (e.g., [74, 98, 33, 51, 10, 206, 25, 226, 166, 68, 39]), thus leading to inaccurate modeling results. These factors evolve over time [33, 132, 137], and are topic dependent [262, 258, 202, 188, 212, 204, 228, 105, 203, 83]. Nevertheless, there are no existing models that consider these dynamic changes.

Lack of integrated community behavior modeling. Similarly, recent analyses have shown that users' personal interest and communities' interest determine both user content and user behavior of multiple types [99, 60, 203, 83, 228, 221, 172]. The previous research works on modeling community behavior however only consider user behavior of a single type [92, 179, 194, 248, 195]. They fail to consider community effect on content and multiple types of behaviors in their models. Moreover, these works assume that a users' personal interest is determined solely by their communities' interest. Such a modeling approach is not practical in the microblogging context where a user is likely to have multiple topical interests not always determined by her communities.

1.2 Research Objectives

In this dissertation, we aim to address the above major shortcomings. Our goal in modeling user behavior in content propagation is to develop models for measuring the three behavioral factors (i.e., user virality, user susceptibility, and content virality). We want to consider inter-relationships of the factors and their dynamics over time and topics. In modeling community behavior, we would like to develop topic models for learning both users' personal interest and user communities' interest from their content and adoption of behaviors of multiple types.

1.2.1 Modeling User Behavior in Content Propagation

Since user virality, user susceptibility and content virality interact with one another as content is propagated, they cannot be measured separately. We therefore propose to simultaneously model them within a common framework. This differentiates our work from existing works that analyze and model these factors independently. The scenario is analogous to the computation of hubs and authorities from a set of links between web pages [117], and users' influence and passivity in social networks [187], except that we now have to consider three instead of two factors.

To deal with the temporal dynamics of the behavioral factors, we propose to incorporate their inter-relationships with a time decay function to develop temporal models for measuring the factors. Moreover, like many other publicly available social media data, users' adoptions and propagations of content items are observable but not their exposure to the items. This poses some challenges in determining the exact propagation path of these items when constructing the models. We therefore design models that do not require knowledge about user - item exposure, nor restrict the number of item adoptions/ propagations by users. Our proposed models allow incremental computation of the factors so as to cope with large data streams at microblogging sites.

Lastly, to address the dependency of the behavioral factors on topics of content, we propose to jointly model the factors specific to topics. Based upon existing methods for discovering topics of microblogging content, we develop models for factorizing observed propagation data to *topic virality*, and *topic-specific user virality* and *topic-specific user susceptibility*. Defined at the topic level, these factors can be used to predict content propagation more effectively.

1.2.2 Modeling Community Behavior

We first address the joint modeling of user content and user behaviors of different types. To work with this multimodal data, we propose to represent both the content and the behaviors as different "bags-of-words". This representation allows user content and user behaviors to be treated differently but modeled in a unified way. We then develop a topic model that is able to simultaneously derive users' topical interests, communities, and the common behaviors of each community.

Finally, we address the joint modeling of users' personal interest and communities' interests. We propose to jointly model the two interests in the same framework where both user content and behaviors are generated by a common set of topics. We would also like to learn users' bias toward her communities in generating content or adopting behaviors. By doing so, we are then able to distinguish the two interests.

1.3 Contributions

Our works in modeling user behavior in content propagation make the following contributions:

- We propose a model, called *Mutual Dependency model* (**md** model), for joint modeling behavioral factors underlying content propagation in microblogging: content virality, user virality, and user susceptibility. We also propose an effective iterative algorithm for learning the model's parameters from data. The model is novel in that it measures the factors simultaneously, exploiting their inter-relationships.

- We adapt the above **md** model to deal with more practical settings where user-content item exposure is not known, and the assumption of single adoption/propagation per item for each user does not hold. We propose new static and new temporal models (**mdu** and **t – mdu** models) for measuring the behavioral factors. Our models incorporate an efficient temporal weighting scheme, considering both the factors' temporal dynamics and their inter-relationships. We also propose an incremental model (**inc**) for efficiently computing the factors in large data streams.

- Lastly, we propose a tensor factorization framework, called **V2S** framework, for attributing the content propagation behavior to topic virality and topic-specific user virality and susceptibility. Based on the framework, we develop two factorization methods: *Numerical Factorization Method* and *Probabilistic Factorization Method* to simultaneously derive topic virality vector, user-topic virality matrix, and user-topic susceptibility matrix. With the learnt vector and matrices, we are able to effectively predict the propagation of future content. This framework and its two factorization methods are novel as the previous state-of-the-art content propagation prediction methods require more user activity data (some even not observed) and are only applicable to the propagation of the current content.

In modeling community behavior, our contributions are:

- We propose the *Community Behavior and Sentiment* (**CBS**) topic model, a probabilistic graphical model, to derive the user communities in microblogging networks based on the sentiments they express on their generated content and behaviors they adopt. As a topic model, **CBS** can uncover hidden topics and derive user topic distribution. In addition, our model associates topic-specific sentiments and behaviors with each user community. Notably, **CBS** has a general framework incorporating multiple types of behaviors simultaneously. This makes **CBS** novel compared with the existing works on mining user behavior that only consider a single type of behavior.

- Finally, to derive and differentiate between personal and community interests, we propose the *Generalized Behavior-Topic* (**GBT**) model for simultaneously modeling community topics and users' topical interest in microblogging data. **GBT** considers multiple topical communities with different topical interests while learning the personal topics of each user and her dependence on communities to generate both content and behaviors. This differentiates **GBT** from other previous works that consider either one community only or user content data only. **GBT** also distinguishes itself from other earlier ones by modeling multiple types of behaviors together.

1.4 Dissertation Structure

The remaining part of this dissertation is as follows. We first review related works in Chapter 2. We then present our works on modeling user behavior in content propagation in Part I. This part includes Chapters 3, 4, and 5. Chapter 3 presents the **md** model for static modeling of content virality, user virality, and user susceptibility. Chapter 4 presents our **mdu** and **t-mdu** models for temporal and online modeling of these virality and susceptibility factors. The **V2S** framework and its derived factorization models for modeling topic-specific virality and susceptibility factors are presented in Chapter 5. Next, Part II describes our works on modeling community behavior. This part includes Chapters 6 and 7. Chapter 6 describes the **CBS** model for simultaneously deriving the community of each user, and the common behaviors and common topic-specific sentiment of each community. In Chapter 7, we present the **GBT** model for deriving and distinguishing personal and community interests using both user content and user behavior. Finally, we conclude this dissertation and discuss some directions for the future work in Chapter 8.

Chapter 2

Related Works

In this chapter, we survey previous literature that are closely related to this dissertation and highlight the differences between our works and the existing ones. We first review studies on user content and community analysis in social networks. Next, we focus on reviewing prior works on mining microblogging user behavior. These include (i) empirical studies on finding patterns of microblogging users in adopting behavior(s), and (ii) methods for identifying and measuring factors underlying users' behavior adoptions.

It is important to note that the study of user behavior and user preference has a long history in social sciences, economics, epidemiology, and computer science. For example, research in social sciences identifies several psychological and social factors affecting how people behave in different social settings, e.g., [121, 142, 183, 184, 182]. Economists study the effects of social, cognitive, and emotional factors on people's economic decisions such as buying a new product or selecting a service, e.g., [24, 186, 170, 43]. Similar research were conducted in epidemiological studies for modeling disease and virus propagation, e.g., [15, 8]. In computer science, user behavior was first studied by human computer interaction researcher to enhance the user experience when working with computers as well as other computing devices [72, 106]. These works were however mostly conducted at small scale. Only in recent years, user behavior and user preference was are studied at large scale using data mining approach.

2.1 User Content Analysis and Community Analysis

2.1.1 User Content Analysis

The analysis of user content is often conducted using topic models. Deerwester *et al.* [54] first proposed LSI model to discover topics of a document corpus by performing spectral analysis on its document-term matrix. Hofmann [88] then developed PLSA model, which is the probabilistic interpretation of LSI. Later, Blei *et al.* [27] proposed LDA model, the bayesian version of PLSA which is hugely popular. In LDA, each document is modeled as a bag-of-words with a multinomial distribution over topics, and each topic has a multinomial distribution over words. Both the documents' topic distributions and the topics' word distributions are assumed to have Dirichlet priors.

Blei *et al.*'s work has triggered a number of works on LDA-like models for mining user interest in social networks based on user content. For example, Tang *et al.* [210] developed a model for discovering researchers' interest from their published papers. Krestel *et al.* [120] applied LDA to a tag recommender system that suggests tags for documents to be tagged. Yano *et al.* [246] developed a topic model for predicting bloggers' comments.

In microblogging, Michelson *et al.* [154] empirically analyzed users' topical interest by examining named entities mentioned in their tweets. Hong *et al.* [92] conducted an empirical study on different ways of performing topic modeling on tweets using the original LDA model [27] and Author-Topic model [190]. They found that the topics learnt from documents formed by aggregating tweets by user could help in user profiling. Similarly, Mehrotra *et al.*

investigated ways of forming documents from tweets in order to improve the performance of LDA model [152]. They found that grouping tweets by hashtag could lead to an improvement in quality of the learnt topics. Ramage *et al.* [179] further proposed to use Labeled LDA model [180] to model topics of tweets where each tweet is labeled based on its linguistic elements (e.g., hashtags, emoticons, and question marks, etc.). Zhao *et al.* proposed TwitterLDA model, a variant of LDA designed specially for tweets, in which: (i) documents are formed by aggregating tweets of the same users; (ii) a single background topic is assumed; (iii) each tweet has only one topic shared by all words of the tweet; and (iv), each word in a tweet is generated from either the background topic or the tweet's topic. Based on the same assumptions, Qiu *et al.* [177] proposed to model topics of tweets using both the tweets' content and the types of their associated behavior types (i.e., either a tweet is a *(original) tweet* or *retweet*, etc.)

There are other works extending topic models beyond content analysis in Twitter. Nallapati *et al.* [158] proposed to jointly model Twitter users' tweets and their follow links. Lim *et al.* [131] and Tan *et al.* [209] incorporated sentiment of tweets into LDA. Wang *et al.* [224] proposed to regularize LDA by user network information. Vosecky *et al.* proposed to jointly model multiple types of named entities embedded in tweets [218, 219]. Yan *et al.* [234] and Cheng *et al.* [41] proposed to model the generation of co-occurrence of word pairs instead of modeling the occurrence individual words. Lin *et al.* proposed to exploit the sparsity in both topic distributions and topic-word distributions [133] for modeling topics of tweets. Yang *et al.* proposed a classification approach to assign tweets to pre-defined topics [242].

Our proposed models for modeling user content and behavior are also designed based on LDA and TwitterLDA. Our research in this dissertation is however different in several ways. Firstly, users' topical interests determine both their content and behaviors. Existing works however model topics of either user content or user behavior. Hence, user interests are modeled in a less-than-optimal manner. We overcome this by considering both user content and user behavior in a common framework, learning user interest from both data sources. Secondly, these works do not consider user community in learning user interest. On the other hand, we propose to simultaneously model both user communities and user interests.

2.1.2 Community Analysis

In social networks, communities are formed by users developing social ties with other users, or users sharing common interest with others, or by both. This results in two main types of communities, i.e., *social* and *topical*, and their *hybrid* [174, 71]. A social community has more dense social links and interactions among the community users, even when the topical interests of users in each community may vary significantly. On the other hand, a topical community may not have many social links and interactions among its members, but these members share common topical interests. Ding *et al.* [57] conducted an empirical study showing that social communities can be significantly different from topical communities from the same network.

Most of the early works on community analysis focus on finding social communities. Social communities are detected solely based on user links. Lately, researchers have proposed to detect user communities using both social network and user attributes, user content, and user interactions (including links and exchanged messages), which result in topical and hybrid communities. We summarize all these works in Table 2.1 in two dimensions. The first dimension is community type which is: either *"social"* or *"topical and hybrid"*. The second dimension is the overlapping constraint among the communities. Overlapping communities are ones that share common members.

Non-overlapping social community detection. Newman *et al.* proposed to discover social communities by finding a network partition that max-

Table 2.1: Related works on community analysis

	Social	Topical and Hybrid
Non-overlapping	Newman *et al.* [160]	Zhou *et al.* [261]
	Newman *et al.* [163]	Akoglu *et al.* [4]
	Newman *et al.* [161]	Ruan *et al.* [193]
	Clauset *et al.* [44]	
	White *et al.* [227]	
	Newman *et al.* [162]	
	Ruan *et al.* [192]	
	Andersen *et al.* [6]	
	Andersen *et al.* [7]	
	Kloumann *et al.* [118]	
	Raghavan *et al.* [178]	
	Ugan *et al.* [215]	
	Holland *et al.* [89]	
	Karrer *et al.* [110]	
Overlapping	Palla *et al.* [164]	McCallum *et al.* [151]
	Lee *et. al* [126]	Zhou *et al.* [260]
	Wang *et. al* [223]	Ramnath *et al.* [19]
	Psorakis *et. al* [175]	Sachan *et al.* [194]
	Yang *et. al* [237]	Liu *et al.* [141]
	Yang *et. al* [239]	Yin *et al.* [248]
	Airoldi *et al.* [3]	Sachan *et al.* [195]
	Gyenge *et al.* [76]	Xu *et al.* [232]
	Ahn *et al.* [2]	Kim *et al.* [113]
	Fortunato *et al.* [64]	Yang *et al.* [238]

imizes a measure of “compactness” in community structure called *modularity* [160, 163, 161, 44]. White *et al.* [227], Newman *et al.* [162], and Ruan *et al.* [192]then proposed different graph spectral-based methods for the modularity optimization. However, Fortunato *et al.*[65] showed that modularity fails to detect social communities when the number of network links is relatively much larger than the communities' size. Based on the classification approach, Andersen *et al.* [6, 7] and Kloumann *et al.* [118] proposed random walk-based methods for detecting social communities by expanding from the sets of community seed users. Similarly, Raghavan *et al.* [178] and Ugander *et al.* [215] proposed label propagation methods for assigning community label for users. Lastly, adopting the Bayesian probabilistic approach, Holland *et al.* [89] and Karrer *et al.* [110] proposed block-based generative models which assume that each user belongs to a *block* (i.e., social community), and the links among users are the result of the blocks' interactions. The users' communities are then determined by fitting the models' parameters using the observed links.

Overlapping social community detection. Palla *et al.* [164] showed that overlaps among social communities can be significant. Shen *et al.* [200] and Lee *et. al* [126] then proposed methods for detecting clique-like overlapping communities. Wang *et al.* [223], Psorakis *et al.* [175], and Yang *et al.* [237, 239] investigated the application of matrix factorization methods for general overlapping community detection. Based on the Bayesian probabilistic approach, Airoldi *et al.* proposed a statistical mixed membership model [3] for generating network links from user blocks' interactions, wherein each user has a multinomial distribution over the blocks. Gyenge *et al.* [76] later proposed a LDA-like model which is more suitable for sparse networks. Lastly, Ahn *et al.* [2] and Fortunato *et al.* [64] proposed to discover social communities by performing clustering on network links instead of nodes.

In our works, we also use social network and user interaction for uncovering both non-overlapping and overlapping communities. We however do not explicitly model the social links and pairwise interactions among users. Instead, we model both the links and interactions as user behaviors, such as following other users, or mentioning other users in tweets, etc..

Non-overlapping topical and hybrid community detection. Zhou *et al.* [261] developed an unified random walk from users' network and their attributes to measure pairwise user similarity for user clustering by K-Medoids algorithm. Similarly, Ruan *et al.* [193] used the user similarity measure computed from their content and links. Akoglu *et al.* [4] proposed a heuristic algorithm for rearranging and assigning users to non-overlapping communities such that the cost for a lossless compression of the network and user attribute matrices is minimized.

Overlapping topical and hybrid community detection. Bayesian learning researchers have developed LDA-like and mixed membership-based methods for joint modeling user community and topics based on different data sources. For example, McCallum *et al.* [151], Zhou *et al.* [260], Ramnath *et al.* [19], and Sachan *et al.* [194] used messages exchanged among the users; Liu *et al.* [141], Yin *et al.* [248], and Sachan *et al.* [195] used network and user content; Xu *et al.* [232], Kim*et al.* [113] and Yang *et al.* [238] used network and attributes. In these works, each user has a multinomial distributions over the communities, and each of the communities has its own preferences in generating content, user attributes, or interactions with other communities. User communities and their preferences are then learnt simultaneously by fitting the models' parameters using observed data.

Our works also focus on mining topical communities, but differentiating between users' personal and communities' interests. In all the aforementioned works, the topical interest of each community refers to the most common topics shared by users within a social community, and hence may not uniquely characterize the community (e.g., two communities may have some common topical interest). Moreover, these works fail to differentiate users' personal interest from that of their communities. They assume that a user's topical interest is determined purely based on her communities' interests. This assumption is not practical in the microblogging context since microblogging users cover a vast range of interest topics, which are not always determined by their topical communities. Without distinguishing the two kind of interests, the previous models would not be able to describe the users' personal interest very accurately. We address this shortcoming by jointly model user and community interests, as well as the bias of each user toward her communities in generating content and adopting behavior.

Table 2.2: Related works on microblogging user behavior analysis and mining

	Behavior type			
	Relationship	Communication	Propagation	Linguistic
Empirical research	Bernado *et al.* [96] Cha *et al.* [33] Romero *et al.* [189] Golder *et al.* [70] Kwak *et al.* [123, 122] Wu *et al.* [228] Feller *et al.* [60] Kivran-Swaine *et al.* [116] Kwak *et. al* [124] Xu *et al.* [230] Hutto *et al.* [97] Antoniades *et al.* [9] Myers *et al.* [155]	Java *et al.* [103] Honeycutt *et al.* [90] Sousa *et al.* [201] Conover *et al.* [47] Cheng *et al.* [40] Bak *et al.* [16] Macskassy *et al.* [148] Kim *et al.* [114] Comarela *et al.* [46] Schantl *et al.* [198] Purohit *et al.* [176] Bak *et al.* [17] Garcia-Gavilanes *et al.* [67]	Lerman *et al.*[129], Nagarajan *et al.* [157] Suh *et al.* [204] Wu *et al.* [228], Romero *et. a.* [188] Macskassy *et al.* [149] Conover *et al.* [47] Petrovi *et al.* [171] Hansen *et al.* [78] Starbird *et al.* [202], Xu *et al.* [231] Stieglitz *et al.* [203] Wang *et al.* [221] Zhiming *et al.* [259] Sun *et al.* [205], Hoang *et al.* [83] Tan *et al.* [208]	Yang *et al.* [236] Cunha *et al.* [49] Zanzotto *et al.* [251] Zappavigna *et al.* [252] Lehmann *et al.* [128] Yang *et al.* [241] Kooti *et al.* [119] Lin *et al.* [135] Dong *et al.* [59] Cunha *et al.* [50]
Modeling research	Hannon *et al.* [77] Kim *et al.* [115] Yin *et al.* [247] Hopcroft *et al.* [95] Lou *et al.* [143] Barbieri *et al.* [23]	Ritter *et al.* [185] Chen *et al.*[35] Chelmis *et al.* [34] Artzi *et al.* [12]	Weng *et al.* [225] Yang *et al.* [245] Liu *et al.* [140], Dabeer *et al.* [51] Hong *et al.* [91], Peng *et al.* [167] Uysal *et al.* [216], Romero *et al.* [187] Yan *et al.* [233], Diego *et al.* [196] Chen *et al.* [36], Artzi *et al.* [12] Achananuparp *et al.* [1] Jenders *et al.* [104], Zhang *et al.* [254] Hong *et al.* [94], Feng *et al.* [61] Luo *et al.* [144], Pan *et al.* [165] Can *et al.* [31], Yang *et al.* [244] Bian *et al.* [26], Liu *et al.* [138] Gao *et al.* [66], Zhang [256] Zhang *et al.* [255], Lee *et al.* [127]	Zangerle *et al.* [250] Tsur *et al.* [212] Ma *et al.* [146] Khabiri *et al.* [112] Ding *et al.* [58] Kywe *et al.* [125] Kamath *et al.* [109] Feng *et al.* [62] Ma *et al.* [147]

2.2 Empirical Research on Microblogging User Behavior

In this section, we review related empirical research on microblogging user behavior that motivates our research. We summarize these works in the third row of Table 2.2 which are grouped by behavior type. In each cell, we sort the works in chronological order.

2.2.1 Relationship Behavior

Bernado *et al.* [96] examined the follow network among Twitter users and the interaction among them. They found that the follow relationship is weak in the sense that users have interactions with only a very small proportion of their followers and followees. Similar findings were then replicated by Cha *et al.* [33]. These works suggest that only few declared friends are actual friends.

Kwak *et al.* [123] studied the topological features of the Twitter follower graph. They found that the distribution of followers is highly skewed while the rate of reciprocated ties is much lower, and hence concluded that Twitter is more an information sharing network than a social network. Wu *et al.* [228] and Feller *et al.* [60] showed the existence of homophily in following behavior, i.e., Twitter users are more likely to follow other like-minded users.

Romero *et al.* [189] examined the link formation process in Twitter. They showed that the process behaves like the *transitivity process* in *preferential attachment networks*. This was then confirmed by a human study by Golder *et al.* [70], which shows that transitivity and mutuality are more important than other network structural characteristics for users to form links.

Similarly, Hutto *et al.* [97] observed the formation of Twitter following links to a number of active and highly followed users. They found that, in a long run, a user's content, her interactions with other users, and her ego network's structure have similar effect in getting new follow links. Later, Antoniades *et* *al.* [9] showed that there are retweet behaviors that lead to new follow link formed. In contrast, Kwak *et al.* [122], Kivran-Swaine *et al.* [116], and Xu *et al.* [230] studied the unfollow behaviors. They found that Twitter users frequently unfollow other users, and the major factors for the unfollowing include the non-reciprocity of the relationships and the followees' non-informativeness.

Lastly, Myers *et al.* [155] examined the dynamics of Twitter follow network as a function of the information propagation processes in the network. They found that information diffusion may lead to bursts in both new follow and unfollow behaviors.

2.2.2 Communication Behavior

Java *et al.* [103] first showed that Twitter is often used for personal communication. Honeycutt *et al.* [90] investigated how users used Twitter as a tool for conversation and collaboration. Later, Sousa *et al.* [201] studied the motivation for users to communicate in Twitter. By examining conversations about politics, sports, and religion topics, they found that: (1) in general, Twitter users are socially motivated to communicate with each others; however, (2), users with large ego networks are more topically motivated. Macskassy *et al.* [148] found that Twitter users are not active in communicating with each others, and most of the conversations involve only two users.

Conover *et al.* [47] compared the networks induced from communication (i.e., mention) and propagation (i.e., retweet) behaviors among politics oriented Twitter users. They found that while the retweet network is highly polarized, the mention network is not. They further found that users who use more neutral hashtags are more likely to engage in communication with opposing political communities. Cheng *et al.* [40] showed that users having similar ego network structures are more likely to have reciprocal communication. Recently, Garcia-Gavilanes *et al.* [67] showed that the inter-countries mention network preserves economic, social, and cultural boundaries among the countries.

Schantl *et al.* [198] investigated the factors influencing Twitter users to reply to a message. Their findings suggest that social factors, which describe the strength of relations between users, are more influential than topical factors. This indicates that Twitter users' communication behavior is largely affected by social relations than by topics.

Purohit *et al.* [176] examined the differences between conversational and non-conversational tweets. They showed that there are domain-independent linguistic features that can help to distinguish between the two kinds of tweets.

Kim *et al.* [114] examined emotional transitions, emotional influences among the Twitter conversation partners. They found that conversational partners are more likely either express the same emotion, or tends to respond with a positive emotion. Comarela *et al.* [46] showed that Twitter users are more likely to reply to other moderately active users who had some prior communication with them and recently post some tweets. Lastly, Bak *et al.* [17] showed statistical evidences that frequent conversation leads to high self-disclosure in Twitter users, which in turn leads to longer conversations among them.

2.2.3 Propagation Behavior

Existing empirical studies on propagation behavior of microblogging users consist of (a) works on examining the information flow through propagation behaviors of users, and (b) works on identifying factors that affect the propagation behaviors.

In the first sub-category, Lerman *et al.*[129] first examined the differences between microblogging users' propagation behaviors and those of other social networks, and suggested that information can spread better in microblogging networks. Wu *et al.* [228] then found that the information originating from the media propagates in Twitter in a two-step process: (1) the information passes from media to a intermediate layer of "opinion leaders", and then (2) the information passes from the users in intermediate layer to their followers who are less connected and exposed to the media.

In the second sub-category, researchers have examined effects of user, content, linguistic, and sentiment factors on tweets' retweetability in Twitter.

Nagarajan *et al.* [157] first presented case studies suggesting that tweets' retweetability depends on topics. Suh *et al.* [204] showed that the existence of URLs and hashtags in a tweet have strong correlation with its retweetability, while the number of past tweets of a user does not affect the user's likelihood of retweeting. Petrovi *et al.* [171] repeated these findings and also found that authors' authoritativeness has positive effects on a tweet's retweetability. Later, Zhiming *et al.* [259] also showed the positive effects of authority features such as trustworthiness, expertise, and attractiveness.

Using a Wikipedia-based approach, Macskassy *et al.* [149] examined the similarity between Twitter users' retweets and their own (original) tweets. They found a very low similarity between the two kinds of tweets though this similarity plays a significant role in users' retweeting behavior. Similarly, Xu *et al.* [231] found that content factors are less important in making users to retweet than user and interaction factors.

Romero *et. a.* [188] later found that Twitter users exhibit different behaviors when propagating hashtags of different types and topics. They characterized the behavior differences between users by the probability that a user adopts a hashtag after repeated exposure to the hashtag. In particular, they found that the adoption of politically controversial hashtags is especially affected by multiple repeated exposures. On the other hand, such repeated exposures have a much less effect on the adoption of conversational hashtags.

Similarly, Hansen *et al.* [78] examined the effects of sentiment factors. They showed that negative news and positive non-news tweets are more likely be retweeted. Wang *et al.* [221] then examined both sentiment and linguistic factors. They found that simple and concise tweets are more likely to be retweeted. Lastly, Tan *et al.* [208] conducted an author-controlled experiments to investigate the effect of wording. They showed that one may get her tweets retweeted better by adding more information, using language that is similar to both the community norms and her prior messages, and mimicking news headlines.

In the politics domain, Starbird *et al.* [202] first observed highly retweeted users in a mass political event and showcased the differences in getting retweets between users attending the event and those who report the event remotely. Conover *et. al* [47] examined the retweet network among Twitter users (i.e., edges are drawn from a user to other users she retweets). They found that the network is highly politically polarized, i.e., users tend to retweet more from other users sharing the same political affiliation. Stieglitz *et al.* [203] examined a set of political tweets and found that there is a positive correlation between the number of sentiment words in a tweet and the tweet's retweetability. Hoang *et al.* [83] further investigated the effect of sentiment and community factors across different topics.

The above works suggest that the propagation of content in microblogging is jointly determined by user factors and content factors. Next, the factors are topic specific and dynamic. We are therefore motivated to investigate the inter-relationships among the factors to design static and temporal models for better modeling of the propagation of content in microblogging.

2.2.4 Linguistic Behavior

Yang *et al.* [236] first studied the temporal patterns of Twitter crowd in adopting hashtags. They found 6 typical adoption patterns of the most adopted hashtags. Cunha *et al.* [49] later found that the adoption of hashtags behaves like a preferential attachment process: more frequently adopted hashtags will more likely to be adopted. Kooti *et al.* [119] studied the evolution of conventional words for communication objective in Twitter. Lehmann *et al.* [128] and Lin *et al.* [135] then conducted similar research for hashtags.

Zappavigna *et al.* [252] showed that instead of using hashtags to capture topics in tweets, microblogging users use hashtags for many other purposes including personalized bookmarking and named entity markup. Similarly, Yang *et al.* [241] showed the evidences that hashtags have dual role: as content bookmarking, and as community membership indicator.

Dong *et al.* [59] examined the differences in biography of United States (US) and Singapore (SG) Twitter users. They found that US Twitter users were far more likely to self-disclose than SG users. Lastly, Cunha *et al.* [50] examined the gender differences of Twitter users in adopting hashtags. They found that, when expressing attitude toward politicians, male users use more imperative verbal forms in hashtags, while female users tend to use more declarative forms.

To summarize, the empirical studies showed that there are both individual factors and community factors affecting the behavior adoptions of microblogging users. The effects of the two factors are consistent across different types of user behavior. However they are not separately observed. This motivates us to also jointly model the two factor in a common framework that considers multiple types of user behavior.

2.3 Modeling Research on Microblogging User Behavior

We now review related works on modeling microblogging user behavior. Again, we list them in the bottom row of Table 2.2.

2.3.1 Relationship Behavior

Most works on modeling relationship behavior in microblogging is formulated as a recommendation problem. Hannon *et al.* [77] proposed a collaborative filtering based model for recommending users to follow in Twitter. Their model recommends a user to follow other users who (1) have similar (past) tweets, and (2) have similar followees and followers. Based on the same approach, Kim *et al.* [115] later proposed a topic model for recommending followee for a given user based on the user's interest and her current followees' interest. On the other hand, Yin *et al.* [247] proposed a personalized followee recommendation model that is purely based the on the target user's ego network structure. Using social theories of structural balance and homophily, Hopcroft *et al.* [95, 143] proposed a factor graph model for predicting reciprocal link and triadic closure formation in Twitter. Recently, Barbieri *et al.* [23] proposed a topic model that investigating both user susceptibility and authoritativeness for link prediction.

2.3.2 Communication behavior

Ritter *et al.* [185] first proposed unsupervised model for detecting dialogue structure in Twitter based on clustering of raw utterances. Chen *et al.*[35] later proposed to exploit more thread length, topic, and social tie strength for finding interesting conversations in Twitter. Similarly, Chelmis *et al.* [34] proposed a classification-based method for predicting communication links in Twitter. Lastly, Artzi *et al.* [12] created a set of features for predicting if a given tweet will receive responses, including replies and retweets.

2.3.3 Propagation Behavior

Prior works on modeling propagation behavior in microblogging can be classified into two sub-categories: (1) works on finding influential and/or susceptible users; and (2) works on predicting future propagation.

In the first sub-category, Weng *et al.* [225] first proposed a topical pagerank based measure for finding influential users in Twitter. Liu *et al.* [140] then proposed a generative graphical model which utilizes the heterogeneous link information and the textual content associated with each node in the network to mine pairwise influence among Twitter users at topic level. Romero *et al.* [187] proposed an inverse reinforcement model for simultaneously finding influential and passive users in Twitter based on the users' propagating behaviors. Similarly, Achananuparp *et al.* [1] proposed a mutual dependency model for identifying originating and promoting users. Diego *et al.* [196] went beyond finding influential users by proposing a page-rank based algorithm for finding trendsetters who propagated content items to influential users. Lastly, Yang *et al.* [244] proposed topic model for identifying users' social roles in adopting propagation behaviors.

The second sub-category includes works on: (a) retweet prediction - to if a user retweets a tweet; and (b) viral tweet prediction - to predict if a tweet will be highly retweeted.

Yang *et al.* [245] first proposed a factor graph model for retweet prediction based on the given tweet's retweet trace. Dabeer *et al.* [51] proposed a framework to measure probability that a tweet is retweeted purely based on the followers of the user posting the tweet. Peng *et al.* [167] proposed a conditional random field model to predict retweet using content, network, temporal features. Zhang *et al.* [254, 255] developed a temporal model for measuring how a user influenced by other users in her ego-network, and made use of the measure to predict if the user retweets a tweet before a given time.

Other researchers considered retweet prediction as a recommendation task. Uysal *et al.* [216] first proposed a filtering model that both recommends tweets for a target user to retweet, and recommend users who more likely to retweet a given tweet. Yan *et al.* [233] random walk based for tweet recommendation. Chen *et al.* [36] proposed a personalized collaborative ranking method that investigates content and social features for recommending tweets for a user. Hong *et al.* [94] and Feng *et al.* [61] developed factorization models for learning users' preference in making retweets. Luo *et al.* [144] proposed a learning to rank based method for recommending users to a tweet. Pan *et al.* [165] proposed to integrate the advantages of collaborative filtering and the characteristics of propagation processes in personalized tweet recommendation. Liu *et al.* [138] and Zhang [256] proposed both topic models for learning users' temporal preference in retweeting. Lastly, Lee *et al.* [127] proposed models for recommending out-ego-network users for a tweet.

Hong *et al.* [91], Jenders *et al.* [104], and Can *et al.* [31] created different sets of features for viral tweet prediction. Bian *et al.* [26] later proposed a multi-task transfer learning model for predicting both viral tweets and users will likely to retweet the tweets. Recently, Gao *et al.* [66] developed a reinforced Poisson process for predicting the number of retweets for a given tweet.

Although propagation behavior in microblogging is widely studied, prior works mentioned above model the underlying user and item factors independently from others, ignoring inter-dependencies among the factors, which are shown in empirical research. Moreover, empirical studies have also shown that the user and item factors change rapidly. Nevertheless, there is no existing model that considers these temporal dynamics. We address these issues by investigating the inter-dependencies to develop both static and temporal models for simultaneously measuring user and item factors. Lastly, existing models measure users' virality and susceptibility independent of topics. Such topic-independent approach can lead to inaccurate modeling results. We address this problem by developing a framework that allows us to simultaneously measure the factors at topic level.

2.3.4 Linguistic Behavior

Most of previous works on modeling linguistic behavior in microblogging focus on: (1) linguistic elements (e.g., hashtags) recommendation - to recommend linguistic elements for a user to adopt in her tweets, and (2) viral hashtag prediction - to predict if a hashtag will be frequently used.

In the first sub-category, Zangerle *et al.* [250] proposed to recommend hashtags to users purely based on the content of the tweet being posted as they assumes that the primary purpose of the hashtags is to categorize the tweets and facilitate the search. Khabiri *et al.* [112] approached the problem as a link prediction task on the graph among hashtags and tweets. Kywe *et al.* [125] proposed a simple neighbourhood-based model for hashtag recommendation where the neighbours are either similar users or similar tweets. Ding *et al.* [58] proposed a topical translation model for simultaneously modeling tweet content and hashtags, assuming that the content and hashtag(s) of a tweet are talking about the same theme but written in different languages. Feng *et al.* [62] developed a learning to rank model for personalized hashtag recommendation. Lastly, Ma *et al.* [147] proposed a topic model joint modeling of both the hashtags and the tweets' content.

In the second sub-category, Tsur *et al.* [212] proposed a regression model for predicting hashtag popularity based on a rich set of content, social, and temporal features. Similarly, Ma *et al.* [146] made use of content and social features to predict if a hashtag will be viral the next day.

In summary, despite a number of research on analysis and mining of user behavior in microblogging, the existing works consider only a single type of user behavior, and/or do not use content while modeling user behavior. Taking a unified approach, we propose to model both user content and user behavior of different types, sharing a common set of latent topics. This approach allows us to better learn users' interest keeping the topics consistent across user content and user behavior. It also allows one to make inference of user behavior using the content, and vice versa. Lastly, previous works fail to capture the community effects on behavior adoptions of users, as shown in empirical research. We therefore propose to learn topical communities, in addition to users' personal topical interest and their dependence on the communities when generating content and adopting behavior.

Part I

Modeling User Behavior in Content Propagation

Chapter 3

Virality and Susceptibility in Content Propagation

In this chapter, we study the problem of modeling both content and user factors underlying content propagation behaviors of microblogging users. We identify *user virality*, *user susceptibility* and *content virality* the three behavioral factors driving content propagation. Instead of modeling these factors independently as done in previous research, we propose a model that measures them *simultaneously* considering their mutual dependencies. This chapter is organized as follows. We first introduce the behavioral factors and discuss their inter-dependencies in Section 3.1. We then state our research objectives and highlight our contributions in Section 3.2. Next, we describe some existing models as well as our proposed one in Sections 3.3. Our experiments to evaluate the proposed model on synthetic and real datasets are presented in Section 3.4 and Section 3.5 respectively. Finally, we summarize the chapter in Section 3.6.

3.1 Introduction

Content propagates among microblogging users through their follow links, from followees to followers. In content propagation, a user may decide to adopt a content item when she observes the item adopted by her followees. The propagated content item can be a message, URL, hashtag, or some other unit of information which can be disseminated among users. Consider a Twitter user network shown in Figure 3.1. In this example, users v_1 to v_5 follow users u_1, u_2, and u_3, and the propagated content are $t_1, \cdots, t_{14}$, and t_{15}. When v_1 adopts t_1 which is previously adopted by u_1, we say that t_1 is propagated from u_1 to v_1. At the same time, the hashtag #*edu* is also propagated from u_1 to v_1. In this example, u_1 is the *propagating user*, and the v_1 is known as the *infected user*.

Existing empirical works have shown that there are three important behavioral factors that affect content propagation behaviors of microblogging users, namely: (a) virality of the user propagating the content item, (b) susceptibility of the user infected with the item, and (c) virality of the content item [204, 33, 188, 149, 18, 25, 203, 259]. We call them the ***user virality***, ***user susceptibility*** and ***content virality*** respectively. User virality refers to the ability of a user in propagating content items to other users while user susceptibility refers to the tendency of a user to be infected by items propagated from others. Content virality refers to the tendency of a content item to attract adoptions by many users through propagation. These three behavioral factors are vital to many applications. For example, viral content can be exploited for advertisement and marketing [101]. Viral users can be engaged to dispel rumors or to conduct product campaigning [69]. Finally, non-susceptible users' mentions of events can be regarded to be important [1].

Empirical research in the past have suggested there are inter-dependencies among the three behavioral factors [228, 105, 47, 10, 203, 83]. Most previous works however measure each factor independently from the others, e.g.,[74, 98, 33, 51, 10, 206, 25, 226, 166, 68, 39], thus leading to inaccurate modeling results.

Consider the example in Figure 3.1. Without considering the followers'

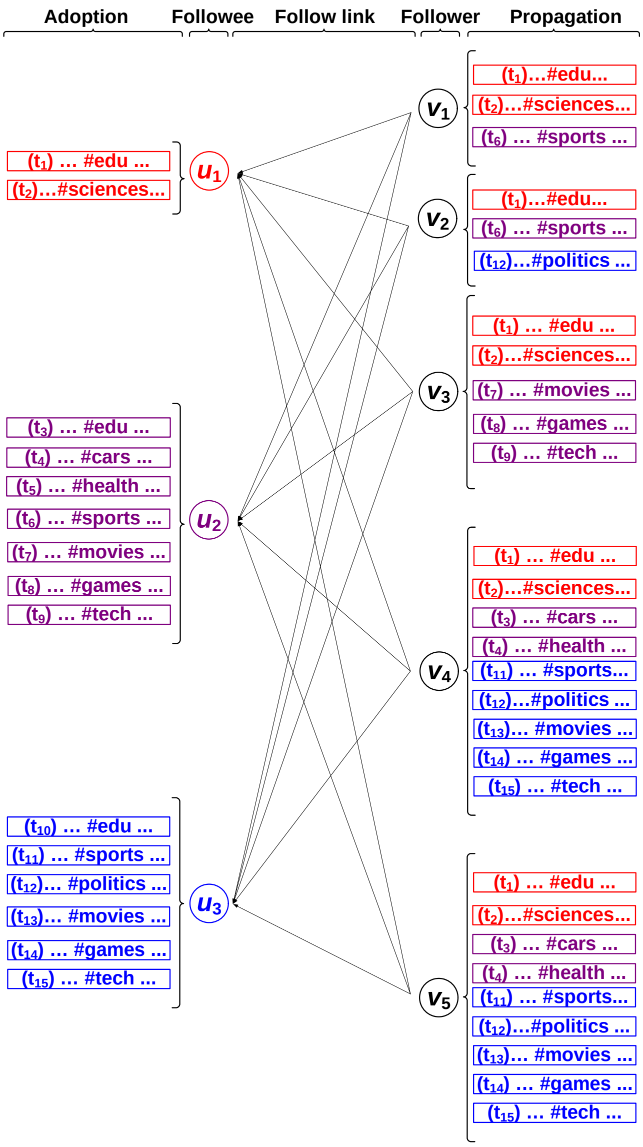


Figure 3.1: Illustrative example of content propagation in microblogging.

susceptibility one may conclude that u_3 is more viral than u_1 since the former gets more propagation (i.e., 11 times) than the latter (i.e., 9 times). However, v_4 and v_5 are observed to be much more susceptible than other followers since v_4 and v_5 adopt all the hashtags propagated by the followees. The same is not observed on other followers. Moreover, u_3 receives propagation mostly by v_4 and v_5 while all u_1's hashtags are propagated to all the followers. Hence, knowing that v_4 is susceptible user lead us to conclude that u_1 is more viral than u_3.

Similarly, without considering the users' virality and susceptibility, one may conclude that #*edu* and #*sports* are equally viral since the former attracts more propagation than the latter (5 and 4 times respectively). However, most of #*edu*'s propagation is due to u_1, a viral user. #*sports* in contrast attracts propagations from all the users adopting it. Hence, it is more reasonable to conclude that #*sports* is more viral than #*edu*.

3.2 Research Objectives and Contributions

In this chapter, we propose to *simultaneously* measure *content virality*, *user virality*, and *user susceptibility* within a common framework. Once a user adopts a content item that is propagated to her by her friends, we conjecture that this adoption may be due to two sets of factors. The first set includes the factors external to the user network, e.g., advertising. The second set consists of internal factors due to virality of the user propagating the item, susceptibility of the infected user, and the virality of the item. Here, for simplicity, our proposed framework has left out the external factors and assumes that the social relationships are identical. Despite this assumption, we still have to address a few challenges as follows.

- The effect of each of the three factors is not explicitly differentiated. They therefore cannot be measured separately.
- There is no ground-truth information for the virality and susceptibility factors. The modeling results are therefore cannot be evaluated using the conventional ground-truth-based evaluation metrics.

To deal with above challenges, we first identify content item adoptions due to the propagation process. We then simultaneously measure the three factors using a model that is built based on the mutual dependencies among the factors. Lastly, we evaluate the model's ability in recovering the pre-defined ground-truth from synthesized propagation data. We also examine the model's performance in predicting hashtags' viral order, compared with other existing models.

It is important to note that the modeling of virality and susceptibility factors is related but not the same as modeling and maximizing information propagation. The latter focuses on (a) deriving the propagation rate [24, 235, 156], (b) predicting the number of users adopting content item(s) in the future [240, 75, 226, 39], and (c) maximizing the number of users adopting item subject to some constraint(s) [111, 38, 37, 28]. In contrast, our work focuses on deriving user and item behavioral factors from the observed propagations. Also, our work is related to but not the same with works on mining more fine-grained factors underlying user virality/ susceptibility and content virality. Empirical studies have shown that there are such factors, for example users' profile characteristics [173, 11, 83, 259] and their networks [78, 100, 18], emotional and linguistic characteristics of items [188, 25, 203, 83, 208]. Calibrating these fine-grained factors is however beyond the scope of this work.

This work improves the state-of-the-art of content propagation. To the best of our knowledge, there has not been any work modeling virality and susceptibility together. Our main contributions in this work are as follows.

- We introduce virality at both the content item and user levels, and introduce user susceptibility as a factor affecting content propagation.
- We propose a novel quantitative modeling framework, called **md** model, which utilizes the mutual dependency between content virality, user virality, and user susceptibility as we measure them from observed data.
- We develop an iterative computation algorithm to compute the scores of content virality, user virality, and user susceptibility.
- We compare and contrast our model with existing models in our experiments and case studies. The experiments are conducted on both

synthetic and real datasets for different purposes of evaluating the models.

- We propose a task to predict retweet order for hashtags in Twitter. The results show that the virality scores assigned to the hashtags using our proposed model can be used to predict the order more accurately than other models.

3.3 Virality and Susceptibility Modeling

3.3.1 Definitions and Assumptions

We first introduce concepts and definitions of content propagation, and state our assumptions. The main notations used in this chapter are shown in Table 3.1.

We represent a set of users $\boldsymbol{\mathcal{U}}$ and their follow relationships $\boldsymbol{\mathcal{E}}$ by a directed graph $G = (\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{E}})$. A directed edge $(v \underset{f}{\rightarrow} u) \in \boldsymbol{\mathcal{E}}$ represents v *follows* u. The time when v starts following u is denoted by $t(v \underset{f}{\rightarrow} u)$. For simplicity, we assume that each follower v follows a followee u once only. We use $\boldsymbol{\mathcal{X}}$ to denote the set of content items, which can be URLs, hashtags, person names, or any other identifiable information entities. We use (u, x) to denote user u adopting content item x, and use $\boldsymbol{\mathcal{X}}(u)$ to denote the set of all items u adopts. In this chapter, we assume that each user u also adopts an item x once only, and denote the time when u adopts x by $t(u, x)$.

Item x is said be exposed to user v if there is at least one followee of v, say u, exposing x to v. The ways in which a user exposes (or introduces) items she adopted to her friends are different in different online social networks. For example, in Twitter, u may introduce a hashtag to her followers every time she posts a tweet containing the hashtag. Note that an item can be exposed to the same user for multiple times. We denote the set of items exposed to v by $\boldsymbol{\mathcal{X}}^{exp}(v)$, the set of users exposing x to v by $\boldsymbol{\mathcal{F}}^{exp}(v, v)$, and the set of users whom x is exposed to by $\boldsymbol{\mathcal{U}}^{exp}(x)$.

We say that u *propagates item* x *to* v (or item x is propagated from u to v) and denote this by $u \overset{x}{\rightarrow} v$ if the following conditions hold:

- u adopts x before v adopts, i.e., $t(u,x) < t(v,x)$, and v adopts x after v follows u, i.e., $t(v,x) > t(v \underset{f}{\rightarrow} u)$; and
- u exposes x to v before v adopts x by at most some time threshold τ

The *time threshold* τ is introduced to determine if v is infected with x from u. Using time threshold to determine the propagation from one user to another has been used in several previous works [5, 52]. When v adopts x at some time point, v may have several of her followee(s) who already adopted x within τ time units ago. In this case, we say that v is propagated by *multiple* followees. Our model allows a user to be propagated by multiple followees and to propagate to multiple followers.

We denote the set of items u propagates to her followers by $\boldsymbol{\mathcal{X}}^{pro}(u)$, and the set of items propagated to v (by her followees) by $\boldsymbol{\mathcal{X}}^{inf}(v)$. That is, $\boldsymbol{\mathcal{X}}^{pro}(u) = \left\{x \in \boldsymbol{\mathcal{X}}(u) : u \overset{x}{\rightarrow} v \text{ for some } v \in \boldsymbol{\mathcal{U}}\right\}$, and $\boldsymbol{\mathcal{X}}^{inf}(v) = \left\{x \in \boldsymbol{\mathcal{X}}(v) : u \overset{x}{\rightarrow} v \text{ for some } u \in \boldsymbol{\mathcal{U}}\right\}$. Lastly, we denote the set of users whom u propagates x to by $\boldsymbol{\mathcal{F}}^{pro}(u,x)$, and the set of followees of v who propagate x to v by $\boldsymbol{\mathcal{F}}^{inf}(x,v)$.

Since not all users have chances to propagate items to their friends, or to have items introduced to them from the friends, we may not be able to measure virality and susceptibility for every users due to the lack of historical observations. Instead, we identify the subset $\boldsymbol{\mathcal{U}}^{int} \subseteq \boldsymbol{\mathcal{U}}$ including users introducing (exposing) items to their friends, and the subset $\boldsymbol{\mathcal{U}}^{exp} \subseteq \boldsymbol{\mathcal{U}}$ including users having items exposed to them. We then measure virality and susceptibility for users in $\boldsymbol{\mathcal{U}}^{int}$ and in $\boldsymbol{\mathcal{U}}^{exp}$ respectively.

For simplicity, we assume that all users in the network are not aware of the models to be used for measuring their properties related to virality and susceptibility. Hence, users are not expected behave in a way to trick our

Table 3.1: Notations used to describe **md** model.

$\boldsymbol{\mathcal{U}}$	Set of all users
$\boldsymbol{\mathcal{X}}$	Set of all items
$\boldsymbol{\mathcal{U}}(x)$	Set of users adopting item x
$\boldsymbol{\mathcal{X}}(u)$	Set of items adopted by user u
$\boldsymbol{\mathcal{U}}^{int}$	Set of users introducing (exposing) item(s) to their friends
$\boldsymbol{\mathcal{U}}^{exp}$	Set of users exposed to items
$\boldsymbol{\mathcal{U}}^{exp}(x)$	Set of users exposed to item x
$u \overset{x}{\to} v$	User u propagates item x to user v
$\boldsymbol{\mathcal{U}}^{pro}$	Set of users propagating item(s) to their friends
$\boldsymbol{\mathcal{U}}^{inf}$	Set of users infected with items, i.e., infected users
$\boldsymbol{\mathcal{U}}^{pro}(x)$	Set of users who propagate x to > 0 users
$\boldsymbol{\mathcal{X}}^{pro}(u)$	Set of items propagated by u
$\boldsymbol{\mathcal{X}}^{inf}(v)$	Set of items v infected with
$\boldsymbol{\mathcal{X}}^{exp}(v)$	Set of items exposed to v
$\boldsymbol{\mathcal{F}}^{pro}(u,x)$	Set of followers whom u propagates x to
$\boldsymbol{\mathcal{F}}^{inf}(x,v)$	Set of followees who propagate x to v
$\boldsymbol{\mathcal{F}}^{exp}(x,v)$	Set of followees who expose x to v
$I_{mm}(x)$	Virality of content item x as measured by model mm
$V_{mm}(u)$	Virality of user u as measured by model mm
$S_{mm}(v)$	Susceptibility of user v as measured by model mm

proposed models and the network is spam free. This assumption does not always hold and we shall address it in our future research.

3.3.2 Existing Models for Virality and Susceptibility

In the following, we review some existing virality models that have been introduced in previous works. Most of them cover only one of the virality/ susceptibility factors.

Content virality. Two widely used content virality definitions are popularity [87, 28, 226, 30, 199, 75] and viral coefficient [108].

- **Popularity** is defined as the number of users adopting the item.

$$I_p(x) = |\boldsymbol{\mathcal{U}}(x)|/|\boldsymbol{\mathcal{U}}| \text{ for } \forall x \in \boldsymbol{\mathcal{X}} \tag{3.1}$$

- **Viral coefficient** is the average number of friends that a user propagates the item to once she has adopted the item.

$$I_c(x) = \frac{|\boldsymbol{\mathcal{U}}^{exp}(x) \cap \boldsymbol{\mathcal{U}}(x)|}{|\boldsymbol{\mathcal{U}}^{pro}(x)|} \text{ for } \forall x \in \boldsymbol{\mathcal{X}} \tag{3.2}$$

Popularity captures how widely the item is adopted but it does not tell if the adoptions are due to propagation or external influence such as media advertisement. Viral coefficient is defined purely based on the item adoptions due to word-of-mouth. When viral coefficient exceeds 1.0, every user adopting the item is able to get more than one other users adopts the item, making the item propagation viral. Viral coefficient however does not consider user factors.

User virality. The conventional approach to measure user virality is **Fanout**, i.e., the average number of friends she propagates items to [74, 98]. That is,

$$V_f(u) = \frac{\sum_{x \in \boldsymbol{\mathcal{X}}^{pro}(u)} |\boldsymbol{\mathcal{F}}^{pro}(u,x)|}{|\boldsymbol{\mathcal{X}}^{pro}(u)|} \text{ for } \forall u \in \boldsymbol{\mathcal{U}}^{int} \tag{3.3}$$

User susceptibility. Prior works have measured a user's susceptibility by **FanIn**, i.e., the fraction of items the user adopts once she is exposed to them [73, 111, 11].

$$S_f(v) = \frac{|\boldsymbol{\mathcal{X}}^{inf}(v)|}{|\boldsymbol{\mathcal{X}}^{exp}(v)|} \text{ for } \forall v \in \boldsymbol{\mathcal{U}}^{exp} \tag{3.4}$$

3.3.3 Mutual Dependency Model

Content propagation in a network is caused by interactions among users as well as interactions between users and content items being propagated. Given that these interactions occur in a network, one has to consider the mutual dependency relationships among the factors when they are measured. We thus propose **Mutual Dependency Model** (**md** model) to measure content virality, user virality, and user susceptibility simultaneously based on a set of principles that help to distinguish each property from others in propagation.

The three principles are:

- Viral content items (with *high content virality*) can be propagated from less viral users (with *low user virality*) to less susceptible users (with *low user susceptibility*).
- Viral users (with *high user virality*) can propagate less viral content items (with *low content virality*) to less susceptible users (with *low user susceptibility*).
- Susceptible users (with *high user susceptibility*) adopts less viral items (with *low content virality*) introduced to her from less viral users (with *low user virality*).

We operationalize the above three principles into the following content virality, user virality, and user susceptibility definitions.

$$I_{md}(x) = \frac{1}{|\boldsymbol{\mathcal{U}}(x)|} \cdot \sum_{u \in \boldsymbol{\mathcal{U}}^{pro}(x)} \left[(1 - V_{md}(u)) \cdot \sum_{v \in \boldsymbol{\mathcal{F}}^{pro}(u,x)} \frac{1 - S_{md}(v)}{|\boldsymbol{\mathcal{F}}^{inf}(v,x)|} \right] \quad \text{for } \forall x \in \boldsymbol{\mathcal{X}} \quad (3.5)$$

$$V_{md}(u) = \frac{1}{|\boldsymbol{\mathcal{X}}(u)|} \cdot \sum_{x \in \boldsymbol{\mathcal{X}}^{pro}(u)} \left[(1 - I_{md}(x)) \cdot \sum_{v \in \boldsymbol{\mathcal{F}}^{pro}(u,x)} \frac{1 - S_{md}(v)}{|\boldsymbol{\mathcal{F}}^{inf}(x,v)|} \right] \quad \text{for } \forall u \in \boldsymbol{\mathcal{U}}^{pro} \quad (3.6)$$

$$S_{md}(v) = \frac{1}{|\boldsymbol{\mathcal{X}}^{exp}(v)|} \cdot \sum_{x \in \boldsymbol{\mathcal{X}}^{inf}(v)} \left[(1 - I_{md}(x)) \cdot \frac{1}{|\boldsymbol{\mathcal{F}}^{exp}(x,v)|} \cdot \sum_{u \in \boldsymbol{\mathcal{F}}^{inf}(x,v)} (1 - V_{md}(u)) \right] \quad \text{for } \forall v \in \boldsymbol{\mathcal{U}}^{inf} \quad (3.7)$$

In Equations (3.5 - 3.7), the terms $(1 - I_{md}(x))$, $(1 - V_{md}(u))$, and $(1 - S_{md}(u))$ are inverses of content virality of x, user virality of u, and user susceptibility of v respectively. In Equation 3.5, the virality of an item x is derived from the number of adoptions of x by (**a**) its propagating users ($\boldsymbol{\mathcal{U}}^{pro}(x)$) and (**b**) its infected users ($\boldsymbol{\mathcal{F}}^{pro}(u, x)$) after (1) weighting the former by the inverse of their user virality, and (2) weighting the latter by the inverse of their user susceptibility prorated by the number of other users who propagate x to them ($|\boldsymbol{\mathcal{F}}^{inf}(x, v)|$). Given that $I_{md}(x)$ considers adoption count per propagating user, it is an extension of viral coefficient $I_c(x)$.

In Equation 3.6, the virality of a user u is derived from the number of adoptions of items she propagates ($\boldsymbol{\mathcal{X}}^{pro}(u)$) to a set of users ($\boldsymbol{\mathcal{F}}^{pro}(u, x)$) after weighting the items by their inverse content virality and the propagated users by their inverse susceptibility prorated by the number of other users who propagate the same item to them ($|\boldsymbol{\mathcal{F}}^{inf}(x, v)|$). $V_{md}(u)$ is an extension of user virality based on fanout $V_f(u)$ as both consider the number of users whom u propagates item(s) to, i.e., $\boldsymbol{\mathcal{F}}^{pro}(u, x)$.

In Equation 3.7, the susceptibility of a user v is measured by the number of adoptions of items she is exposed ($\boldsymbol{\mathcal{X}}^{exp}(v)$) by a set of users ($\boldsymbol{\mathcal{F}}^{exp}(x, x)$) after weighting the items by their inverse content virality and the average inverse user virality of the exposing users who succeeded in propagation. $S_{md}(v)$ also shares some similarity with $S_f(v)$ in using $\boldsymbol{\mathcal{X}}^{inf}(v)$.

3.3.4 Model Computation

Computing scores in the **md** model is a fixed point problem [253]. We employ the iterative computation method in Algorithm 1 to compute $I_{md}(x)$'s, $V_{md}(u)$'s and $S_{md}(v)$'s. The main idea is to initialize $V_{md}(u)$'s and $S_{md}(u)$'s with some values in $[0, 1]$ so as to compute $I_{md}(x)$'s. The computed $I_{md}(x)$'s and $S_{md}(u)$'s are then used to compute a new set of values for $V_{md}(u)$'s. Next, the new $S_{md}(u)$ values are computed from $I_{md}(x)$'s and $V_{md}(u)$'s. This process repeats until we reach a predefined maximum number of iterations or when the values converge.

We empirically found that the iterative computation method works well for all the synthetic and real datasets (more than 50 of them) in our project. The method always converges to a unique users' virality/ susceptibility scores and items' virality scores, regardless of their initializations. We also found that initializing the scores by the normalized uniform vectors (like in lines 1 - 3 of Algorithm 1) causes the method to converge much faster (less than 20 iterations). Proving the convergence of the method is however elusive and is part of our ongoing research.

Algorithm 1 Iterative computation method for computing content virality, user virality, and user susceptibility

```
 1: $(I_{md}(\cdot), V_{md}(\cdot), V_{md}(\cdot)) \leftarrow (\vec{\mathbf{1}}, \vec{\mathbf{1}}, \vec{\mathbf{1}})$    ▷ *Initialization*
 2: $C \leftarrow (I_{md}(\cdot), V_{md}(\cdot), S_{md}(\cdot))$    ▷ *Normalization*
 3: $(I_{md}(\cdot), V_{md}(\cdot), S_{md}(\cdot)) \leftarrow (I_{md}(\cdot), V_{md}(\cdot), S_{md}(\cdot))/\|C\|$
 4: **for** $k \leftarrow 1$ to $MaxIteration$ **do**    ▷ *Update $I_{md}(\cdot)$, $V_{md}(\cdot)$, and $S_{md}(\cdot)$*
 5:     **for** each $x \in X$ **do**
 6:         Compute $I^{'}(x)$ using Equation 3.5
 7:     **end for**
 8:     **for** each $u \in U$ **do**
 9:         Compute $V^{'}(u)$ using Equation 3.6
10:     **end for**
11:     **for** each $v \in S$ **do**
12:         Compute $S^{'}(v)$ using Equation 3.7
13:     **end for**
14:     $C \leftarrow (I^{'}(\cdot), V^{'}(\cdot), S^{'}(\cdot))$    ▷ *Normalization*
15:     $(I_{md}(\cdot), V_{md}(\cdot), S_{md}(\cdot)) \leftarrow (I^{'}(\cdot), V^{'}(\cdot), S^{'}(\cdot))/\|C\|$
16: **end for**
17: Normalize $I_{md}(\cdot)$, $V_{md}(\cdot)$, and $S_{md}(\cdot)$ to unit length
```

3.4 Experiments on Synthetic Datasets

The first set of experiments is designed to evaluate and compare the different virality models including our proposed **md** model. While some of them have been used in the commercial world, a systematic evaluation has not been conducted due to a lack of an existing dataset containing the ground truth labels of viral content items, viral and susceptible users. We therefore create synthetic datasets with different parameter settings and corresponding ground truths and compare the models' accuracies.

3.4.1 Synthetic Data Generation

We use the following steps to generate a synthetic dataset.

Generating the user network. Given the number of users N, power law degree exponent α, minimum degree d_{min}, and maximum degree d_{max}, we generate a undirected network of users whose degree distribution follows the power law with exponent α as follows.

- Generate the degree distribution of N nodes in the $[d_{min}, d_{max}]$ range following the power law distribution using the *inverse transformation method* [191].
- Generate the links for the N nodes to follow the generated degree distribution using the *Expected Degree Model* [42]. The resultant network has each connected pair of users follow each other.

Generating the ground truth. We designate a small number of users, let say k_u of N, who are randomly chosen from users among the top 10 degree percentile, as viral users. This is to ensure that viral users have sufficient followers to propagate item(s) to. The susceptible users are selected the same way. These users are assigned higher virality/ susceptibility scores that are uniformly drawn from $[1-\beta, 1)$ (with $0 < \beta \leq 0.5$), while the remaining users are assigned virality/ susceptibility scores uniformly in the range $[0, \beta)$. We label k_i of M items to be viral. Similarly, these items have virality scores randomly drawn from $[1 - \beta, 1)$, while the remaining items have scores randomly drawn from $[0, \beta)$.

Generating the items adoptions. We generate item adoptions for each item x over 10 time steps. At each time step, as suggested in [24], the probability that each non-adopter v adopts x is $p + q$ where p is the probability attributed to external influence, and q is the probability attributed to internal influence or propagation.

$$q = \frac{1}{3} \cdot \left[1 - \prod_{u \in \boldsymbol{\mathcal{F}}(v,t)} \big(1 - g_V(u)\big) + g_I(x) + g_S(v) \right] \tag{3.8}$$

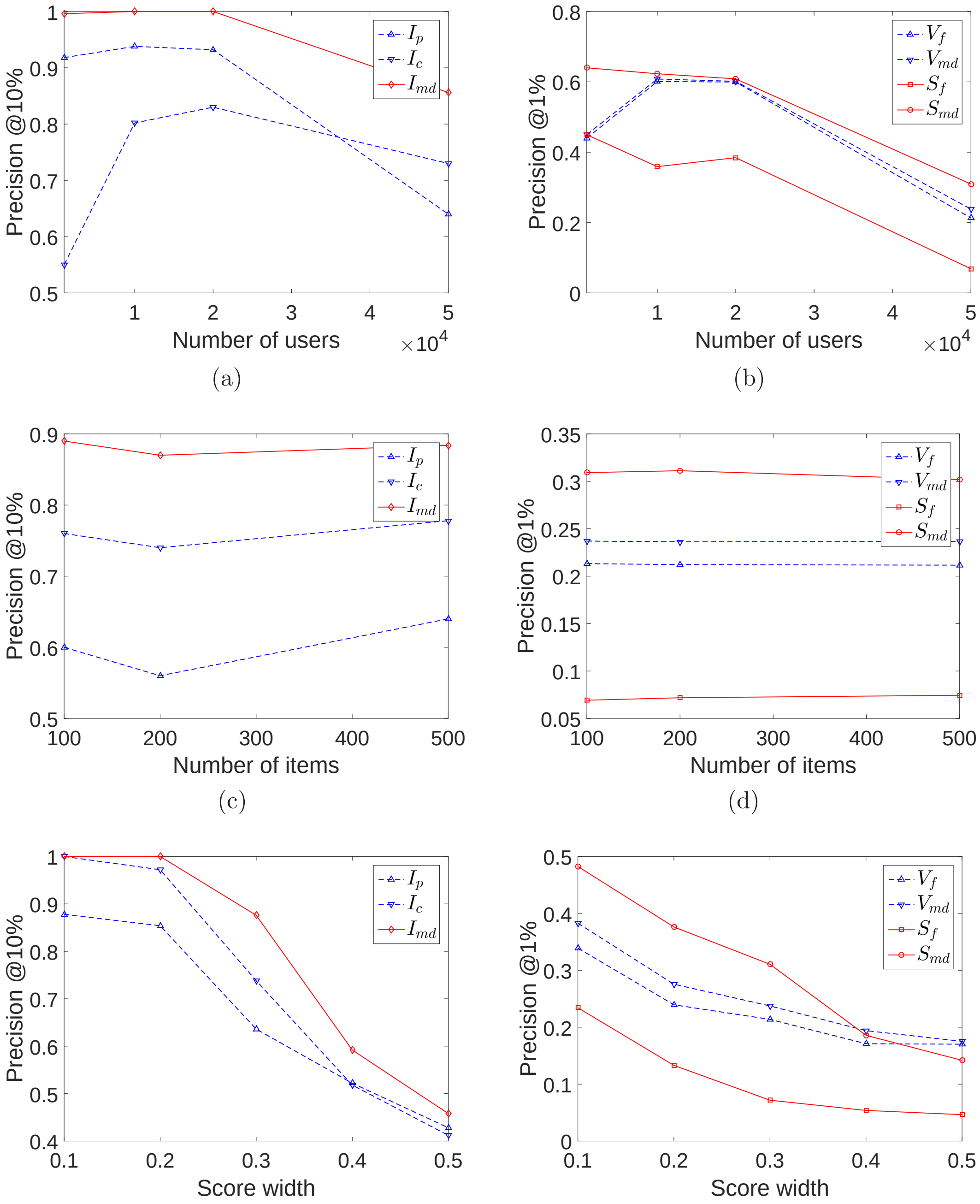
where $\boldsymbol{\mathcal{F}}(v,t)$ is the set of followees of v who adopt x within τ time steps ago, while $g_V(u)$, $g_I(x)$, and $g_S(v)$ are ground-truth user viraliy score of u, content virality score of x, and user susceptibility score of v. In our experiments, we set $\tau = 1$.

We generated networks with different number of users (N is varied from 500 to 5K), number of items (M is varied from 100 to 500), and virality/ susceptibility score width (β is varied from 0.1 to 0.5) while keeping $\alpha = 2.5$, $d_{min} = 1$, $d_{max} = 100$, $k_u = 1\%$ of N, and $k_i = 10\%$ of M. For each parameter setting, we generate 10 instances of item adoptions with p is randomly chosen from $[0.01, 0.05]$ for each item. This range of p is also suggested by experiments on a various type of items reported in Bass [24] and Turk *et. al* [214]. We then compute the virality and susceptibility scores of each dataset instance using different models. For the **md** model, the $MaxIterations$ constant in Algorithm 1 is set to 20.

3.4.2 Results

For each dataset instance, we rank users by their virality (susceptibility) scores produced by a virality model and select the top scored 1% users as the predicted viral (susceptible) users and denote the set by U_v^p (U_s^p). The precision@1% of user virality (susceptibility) is then defined by $\frac{|U_v^p \cap U_v|}{|U_v|}$ ($\frac{|U_s^p \cap U_s|}{|U_s|}$) where U_v and U_s denote the viral users and susceptible users in the ground truth respectively. The precision@10% of content virality is similarly defined.

Figures 3.2 (a) and 3.2 (b) show the precision@10% of content virality and precision@1% of user virality and susceptibility for the different models as we set N = 1000, 10K, 20K and 50K keeping $M = 500$ and $\beta_u = \beta_i = 0.3$. The



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.2: (a), (c), (e): Precision@10% of content virality by varying N, M and score width respectively; (b), (d), and (f): Precision@1% of user virality and susceptibility by varying N, M and score width respectively

figures show that the **md** model outperforms other models, particularly for content virality and user susceptibility. The performance of **md** model in user virality is only slightly better than that of fan-out. All models demonstrate decreasing precision as N increases. They however still outperform the random selection significantly.

Figures 3.2 (c) and 3.2 (d) show the precision@10% of content virality and precision@1% of user virality and susceptibility respectively for the different models as we set M = 100, 200 and 500 keeping $N = 50K$ and $\beta_u = \beta_i = 0.3$. The figures show that the **md** model outperforms all other models. All models demonstrate unchanged precision as M increases.

Figures 3.2 (e) and 3.2 (f) show the precision@10% of content virality and precision@1% of user virality and susceptibility respectively for the different models as we set the score width $\beta_u = \beta_i = 0.1$ to 0.5 keeping $N = 50K$ and $M = 500$. Again, the **md** model outperforms the other. The precision generally falls as we increase the score width. This is expected as larger score width creates ground truth data harder for the models.

3.5 Experiments on a Real Dataset

In this section, we compare the different models using a real Twitter dataset containing tweets published by Singapore-based users during the Singapore's 2011 general election and presidential election. Since the elections are socially interesting events, we expect viral diffusion to exist in the data.

3.5.1 Data Collection and Preprocessing

We first selected a set of of 58 Singapore-based seed users which includes user accounts of the political parties, politicians, political commentators, and bloggers. We then derived the followers and followees of the seed users creating a larger set of 32,138 users who declared themselves to be located in Singapore. We crawled tweets published by the set of users on a daily basis. We collected a set of 30,652,126 tweets published between March and September 2011 for this study. Among those tweets, we have 610,109 retweets.

User network construction. As Twitter does not provide the creation time of follow links, we had to infer the links with timestamp using tweets as suggested in [188]. That is, we created a follow link from user u to user v when u mentions "@v" at least k time in u's tweets. The timestamp of the follow link is thus assigned the timestamp of the k-th tweet of u mentioning "@v". In our experiments, we set $k = 3$.

Item adoption. We use hashtags as items. There have been works suggesting hashtags as the topics of information diffusion in Twitter (e.g., [188, 196]). In this experiment, we consider a user adopts a hashtag when she publishes a tweet containing the hashtag.

Hashtag and user selection. To ensure that we have sufficient observations for each hashtag and each user, we applied the following steps to select hashtags, target users for virality V, and target users for susceptibility S.

- We selected the set of 1000 most popular hashtags
- We selected into V all users adopting at least min_a hashtags in 1000 selected hashtags.
- We selected into S all users having at least min_i selected hashtags introduced to them from users in V.

In our experiment, we set $min_a = min_i = 3$. This gives us $|V| = 12,978$ and $|S| = 11,069$.

Setting the threshold τ. The threshold τ is determined based on the time lag between retweets and their original tweets. We found that the time lag follows a long tail distribution with more than 95% of retweets having timelag within 1 day, and the maximum time lag is 205 days. We therefore set $\tau = 1$ day.

3.5.2 Results

Correlation between different measures. We now examine how the models rank users/ items differently. Table 3.2 shows that the rank correlation coefficient between the I_p and I_c is not high indicating that the popular hashtags are not always well propagated among the hashtag adopters. I_{md} on the other hand is more correlated with both I_p and I_c. I_{md} produces rankings much more similar to I_c than I_p, which is expected as our proposed model tends to give higher ranks to well propagated items.

Table 3.2: Pearson rank correlation of different content virality measures.

	I_p	I_c	I_{md}
I_p	-	0.087	0.145
I_c	-	-	0.733

Similarly, we computed the Pearson rank correlations between V_f and V_{md}, and between S_f and S_{md}. However, since whenever $V_f(u)$ ($S_f(u)$) equals to 0, $V_{md}(u)$ ($S_{md}(u)$) equals to 0, we exclude all such users u from the correlation computation. The correlation coefficient between V_f and V_{md} (respectively between S_f and S_{md}) is 0.87 (respectively 0.97), which indicates that V_{md} and S_{md} are similar but not identical to V_f and S_f.

Comparison of the top-10 viral hashtags. As shown in Table 3.3, the top-10 viral hashtags by different models are quite different. The top-10 by I_p include hashtags related to some big events (e.g., *#sgelections* and *#sgpresidents* for the two elections in Singapore in 2011), or people daily life (e.g., *#nowplaying* for what music people listen to). The top-10 by I_c include mainly hashtags about funny stories and emotion (e.g., *#daveq* and *#overheard*), and those popularized by a single user (e.g., *fakemoe* or *davelimkopi*). As we expected, the top-10 by I_{md} includes more socially and politically related hashtags (e.g., *#crappymediacorptitles* for social problems that are described by phrases similar to the names of some famous song, movie, novels, etc), and for the Singapore's 2011 Presidential Election held on August 27th, 2011 (e.g., *#asksgpresident*).

We further examine *#sgelections*, *#daveq*, and *#crappymediacorptitles*, the three hashtags top ranked for content virality by I_p, I_c, and I_{md} respectively.

Table 3.3: Top 10 viral hashtags rank by different measures.

Rank	I_p	I_c	I_{md}
1	#sgelections	#daveq	#crappymediacorptitles
2	#nowplaying	#everysingaporeandream	#studyinginsingaporeislike
3	#sosingaporean	#ccquotes	#asksgpresident
4	#sgpresident	#overheard	#jobsforgeorgeyeo
5	#fb	#teammilo	#improvefilmtitlesby_ addinginmypants
6	#damnitstrue	#mooncakefestival	#replacesongnameswithcurry
7	#1	#thinkaboutit	#wordspeoplebutcher
8	#justsaying	#sgreans	#chinavssgp
9	#prayforjapan	#kiasu	#yosgpresident
10	#fail	#whyifollowsosingaporean	#notsosingaporean

Table 3.4: Comparison among *#sgelections*, *#daveq*, and *#crappymediacorptitles*

Hashtag	**#sgelections**	**#daveq**	**#crappymediacorptitles**
I_p	6354	223	426
#Infected users	2939	110	333
#Propagating users	1391	5	90
I_c	2.11	22	3.7
I_{md}	0.060	0.062	0.095

As shown in Table 3.4, *#sgelections* has many more adopters than *#daveq* and *#crappymediacorptitles*. However, less than 50% of them adopted the hashtag due to propagation; and less than 25% of them could propagate the hashtag to a small number of followers. This indicates that *#sgelections* is mostly adopted due to some external factors. *#daveq* also has about 50% of the adopters adopting the hashtag due to propagation. However, only a few of them could propagate the hashtag. Furthermore, we found that the propagation of *#daveq* was mostly contributed by a viral user (*fakemoe*). In contrast, more than 75% users adopting *#crappymediacorptitles* adopted the hashtag due to propagation, and about 25% of them could propagate the hashtag. Moreover, we also found that the propagation of *#crappymediacorptitles* was evenly contributed by users diffusing the hashtag. It is thus reasonable to conclude that *#crappymediacorptitles* should be more viral than *#sgelections* and *#daveq*.

Comparison of the top-10 viral users. The top 10 viral users by V_f and V_{md} are identical but not their ranks. They are mainly the social media accounts, portals, bloggers, and fake users.

The two users, *leticiabongnino* and *todayonline*, have significantly different ranks assigned by the two models. *leticiabongnino* is ranked 9th and 7th by V_f and V_{md} respectively, while *todayonline* is ranked 5th and 9th respectively. Although *todayonline* propagated all hashtags it had adopted and has a higher fan-out than *leticiabongnino*, the former could propagate only a few hashtags to many followers. These are viral hashtag related to big social events. On the other hand, *leticiabongnino* could propagate almost all hashtags she had adopted to a large number of followers. Many hashtags that *leticiabongnino* propagated were her own gossip and funny stories that are less likely to be adopted by others. The fact that she could propagate them shows that she has high virality. Therefore, it is reasonable to assign *leticiabongnino* a virality score higher than *todayonline*.

Comparison of the top-10 susceptible users. The top 10 susceptible users by S_f and S_{md} have 6 common users, and their ranks are different. Most of users in the two top-10 are teenages and young adults. Among them are *andyheas79* and *b2utyfulmiley*[1], the two users who have significantly different ranks by the two models. *andyheas79* is ranked 3th and 15th by S_f and S_{md} respectively, while *b2utyfulmiley* is ranked 10th and 8th respectively. We found that the hashtags that *andyheas79* adopted due to diffusion are viral, and were propagated to him by viral users. On the other hand, the hashtags propagated to *b2utyfulmiley* are not very viral, and they came from non-viral users. Therefore, although *andyheas79* has a higher fan-in than *b2utyfulmiley*, it is reasonable to assign *b2utyfulmiley* a higher susceptibility rank.

Summary of Results. Based on the above empirical results on the real dataset, we conclude that the different models produce results that follow our expectation. The **md** model is shown to be more robust as it considers the inter-relationships of all three user and item factors.

[1] This user changed her username to *nanaphew*

3.5.3 Retweet Order Prediction for Hashtags

In this section, we examine the effectiveness of our proposed model when applied in a prediction task. We hypothesize that tweets containing the higher virality hashtags are more likely to be retweeted. We therefore use the virality scores to predict, between a pair of hashtags, which one will have higher retweet likelihood in the near future. To evaluate our prediction model, we conducted the following experiment using the same Singapore-based Twitter dataset, and the same user network constructed from the dataset as described in the previous section.

We divided all tweets into weekly sets based on their published dates. For each week between May and September 2011, we used all tweets published within two weeks before the week as the training set, and used all the tweets published within the week as the test set. We did not examine the first 8 weeks (March and April 2011) as the tweets during this period is mainly used for user network construction. We selected 1000 most popular hashtags in the training set. Virality scores of these hashtags were computed based on diffusion information extracted from the training set. Then, we identified every tuple (u, v, h_1, h_2) of two users, u and v, and two hashtags, h_1 and h_2, that satisfies the following conditions: (a) v follows u; (b) h_1 and h_2 are in the set of 1000 most popular hashtags in the training set (and therefore they had be assigned virality scores); and (c) u posts original tweets using both h_1 and h_2 after v follows u. For each such tuple, we computed the likelihood $l(u, v, h_1)$ (respectively $l(u, v, h_2)$) that v retweets a tweet containing only h_1 (respectively h_2) that appears in the test set, and is originally posted by u after v follows u. If h_1 is more viral than h_2 (as measured by a certain model) and $l(u, v, h_1) > l(u, v, h_2)$, we say that the tuple (u, v, h_1, h_2) supports the prediction model. Obviously, the virality model that gives higher fraction of supporting tuples is better.

Figure 3.3 shows the fraction of tuples of users and hashtags that support

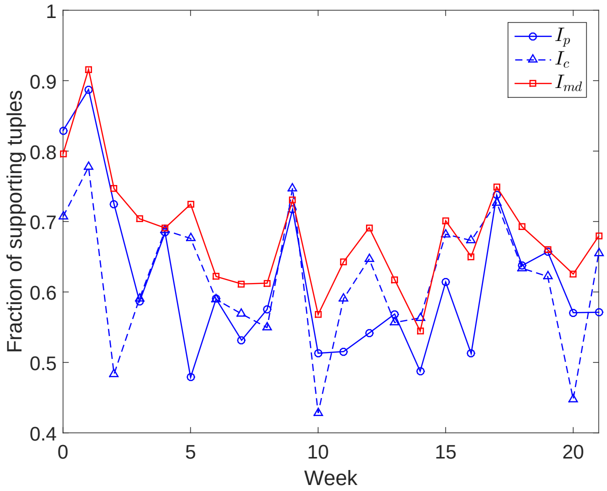


Figure 3.3: Fraction of tuples of users and hashtags supporting the prediction model for every week from May to September 2011

the prediction model for every week from May to September 2011. The average fraction of tuples supporting the prediction model of I_p, I_c, and I_{md} is 0.6, 0.61, and 0.67 respectively. The fractions of all the virality models have three common peaks at week 1, 9, and 17. This is expected as tweets are mainly about big events during these weeks (the general election, the most potential presidential candidates announced their candidacy, and the presidential election respectively). The prediction model based on I_{md} achieves the highest fraction, and also has more stable performance with the faction exceeding 0.6 for almost all the weeks. This shows that our proposed model outperforms other models based on popularity and viral coefficient in this prediction task.

3.6 Chapter Summary

In this chapter, we propose a novel framework to model propagation related user and content behavioral factors. Considering the network effect of users and content items interacting with one another in content propagation, we develop a mutual dependency based model to measure user virality, user susceptibility, and content virality simultaneously. We also develop an algorithm for learning the model's parameters. To evaluate our proposed and other models, we have conducted extensive experiments on both synthetic and real datasets. The experiment results on synthetic datasets have shown that our proposed model generally outperforms the other existing ones. The results on a Twitter dataset have also shown that the proposed model can better approportionate the contributions to content propagation by the different user and content factors properly. The work described in this chapter has appeared in [84].

Chapter 4

Efficient Online Modeling of Virality and Susceptibility in Content Propagation

This chapter presents our work on temporal and online modeling of user virality, user susceptibility, and content virality. We also consider the problem in more realistic problem settings in which user-content item exposure is not observed and each user may have multiple adoptions/ infections with the same content item. This chapter is organized as follows. We first discuss temporal dynamics of the virality and susceptibility factors, and the new problem settings in Section 4.1. We then state our research objectives and summarize our contributions in Section 4.2. Next, we extend existing baseline models for the new problem settings in Section 4.3. We describe our proposed static and temporal models in Section 4.4. Our proposed incremental model is defined in Section 4.5. Our experiments to evaluate the models are presented in Section 4.6. Finally, we conclude the chapter in Section 4.7.

4.1 Motivation

In Chapter 3, we developed the static model **md** for measuring users' virality and susceptibility, and content items' virality, addressing the interrelationships among the factors. Empirical studies however have shown that the factors are temporally dynamic. For example, Lin *et al.* [132] suggested that the set of top viral items change significantly after every hour. Lin *et al.* [134] showed that users' virality and susceptibility change significantly during media events. Nevertheless, to the best of our knowledge, there is no existing works that consider these dynamics. A simple way to address this issue is to apply **md** model to recompute the factors once we get new propagation observations. But **md** model is computationally expensive to be applied in realtime applications with very large data streams.

Moreover, due to the fast evolving of microblogging content, a user can be infected with the same content item multiple times, even from the same propagating user. Consider the propagation scenario shown in Figure 4.1 for example: v_1, v_2, and v_3 follow and receive *tweets* from u_1, u_2, and u_3. When v_1 *retweets* (forwards) the tweet t_1 from u_1, we say that t_1 and the hashtag #*edu* is propagated from u_1 to v_1. In this example case, v_1 is infected with #*edu* two times: once from u_1, and another from u_2. Similarly, v_3 is infected with #*sports* five times: once from u_1, once from u_2, and thrice from u_3. Existing models, including the **md** model, however dismiss the second and subsequent adoptions of the same item [187, 28, 119, 196, 98, 51], thus reducing the accuracy of the modeling results.

Lastly, prior works are based on an important assumption that user-content item exposure is observable. Exposure to an item is the pre-condition of one adopting the item through propagation. In many propagation scenarios, such knowledge is not available. All existing models unfortunately make this assumption to simplify the measurement of the virality and susceptibility factors. They therefore could not be used when only user adoptions can be observed.

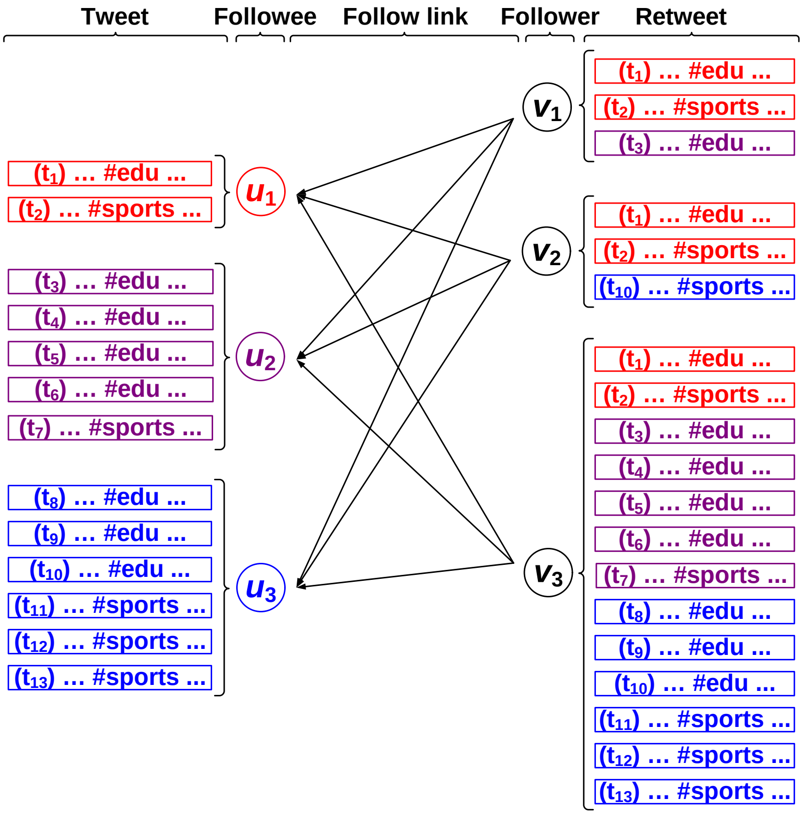


Figure 4.1: Illustrative example of multiple adoptions and infections to with the same content item of microblogging users.

4.2 Research Objective and Contributions

In this chapter, we address the issues mentioned above. Our goal is to develop models for measuring users' virality/ susceptibility and items' virality that (1) consider multiple adoptions and infection of users with the same content item, (2) address the dynamics of the factors, and (3) allow incremental computation of the factors so as to cope with large streams of adoption and propagation data from social media.

The main idea of our approach is to first model the factors using their inter-dependencies without requiring knowledge about what users read nor restricting a user to infect with the same item only once. We then model the dynamics of factors by assigning temporal weights to adoption and propagation instances so that the recent instances are weighted higher than the old ones. This makes the models less bias to the cumulative effects on user and item factors in the whole propagation process. We then consider each propagation instance carrying some amount of work proportional to its temporal weight, and measure the user and item factors by their contributions to the total amount of propagation work aggregated from all the propagation instances. We finally create an incremental method for measuring the user and item factors.

We make the following contributions in this chapter.

- We give a definition of item propagation that can be applied to any general user adoption data as it does not make any assumption of user readership (which is often unobserved), nor the assumption of single infection per item for each user.

- We propose a temporal weighting scheme to assign weight to item adoption and propagation instances. This weighting scheme allows us to give more importance to the recent adoption and propagation instances, as well as to update the weights incrementally.

- We propose both new static and new temporal models for modeling user and item factors in propagation. Our models are built upon the above temporal weighting scheme, considering the temporal dynamics of the factors and their inter-dependencies.

- We also propose an incremental model for efficiently computing the factors from data streams.

- We evaluate our proposed models and other baselines in a large dataset spanning one month. The results show that our proposed models are more intuitive. Our models also outperform the baselines in predicting retweet counts. We also show that our incremental model is more than 10 times faster than the static temporal models, yet obtaining results that are very similar.

4.3 Extension of Existing Models

In this section, we describe static and temporal baselines for user virality/ susceptibility and content virality when user-exposure-to-content item is not observed and users may have multiple adoptions and infections with the same item. We first introduce the main notations used to describe the models. Next, we present the baseline models found in existing works. We then present our extensions of these baselines to consider temporal models. Lastly, we describe our proposed static model and its extension to a temporal model.

4.3.1 Notations

The main notations used in this chapter are shown in Table 4.1. We denote the set of all users and the set of all content items by $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{X}}$ respectively. We use (u, x) to denote an adoption instance wherein u adopts item x, and use (u, x, v) to denote an propagation instance wherein u propagates x to v (implying that v is infected with x). We call u a *propagating user* if u has propagated item(s) to other user(s). Similarly, v is called an *infected user* if v has infected with item(s).

For each user u, there may be more than one (u, x) instances since u may adopt x multiple times. Similarly, for each user v, there may be more than one (u, x, v) instances sharing the same u and x as u may propagate x to v multiple times. The bag of all (u, x) adoption instances is denoted by $\boldsymbol{\mathcal{A}}(u, x)$. The bag of all (u, x, v) propagation instances is denoted by $\boldsymbol{\mathcal{P}}(u, x, v)$. We denote the number of times u adopts x by $\mathbf{a}(u, x)$, and denote the number of times u propagates x to v by $\mathbf{p}(u, x, v)$. That means, $\mathbf{a}(u, x) = |\boldsymbol{\mathcal{A}}(u, x)|$, and $\mathbf{p}(u, x, v) = |\boldsymbol{\mathcal{P}}(u, x, v)|$. In $\mathbf{a}(u, x)$ and $\mathbf{p}(u, x, v)$, a substitution u, x, or v by a dot $(\cdot)$ means that we are taking the summation of $\mathbf{a}(u, x)$ and $\mathbf{p}(u, x, v)$ over all possible values of u, v, and x respectively. For example, $\mathbf{p}(u, x, \cdot)$ is $\sum\limits_{v} \mathbf{p}(u, x, v)$, and $\mathbf{p}(\cdot, x, \cdot)$ is $\sum\limits_{u} \sum\limits_{v} \mathbf{p}(u, x, v)$.

We use t to denote the time step, and use $t(o)$ to denote the time label of the adoption/ propagation instance o. When a set or bag notation has the time subscript t, it denotes the set or the bag whose the elements are observed up to time t. For example, $\mathcal{U}_t^{pro}$ is the set of propagating users up to time t, and $\mathcal{A}_t(u,x)$ is the bag of all (u,x) instances up to time t.

Like in Chapter 3 (see Section 3.3), not all users have propagated or been infected with items, we therefore only measure virality for propagating users, and measure susceptibility for infected users. We denote the set of propagating users by $\mathcal{U}^{pro}$, and denote the set of infected users by $\mathcal{U}^{inf}$. Note that a user may belong to both $\mathcal{U}^{pro}$ and $\mathcal{U}^{inf}$, and $\mathcal{U}^{pro}$,$\mathcal{U}^{inf} \subseteq \mathcal{U}$.

The virality of a propagating user u as derived by a model mm is denoted by $V_{mm}(u)$. Similarly, we use $S_{mm}(v)$ and $I_{mm}(x)$ to denote the susceptibility of infected user v and virality of item x respectively as measured by model mm. Like above, when these notations have time subscript t, it denotes the score as measured at time t. For example, $V_{mm,t}(u)$ is virality of u derived by model mm at time t. $S_{mm,t}(v)$ and $I_{mm,t}(x)$ are defined in the same way.

4.3.2 Static Baseline Models

We now present the baseline models that have been proposed in previous works, and extend them to apply in the new problem settings used in this chapter.

Baselines for user virality. These include **Fan-out** [74, 98] and **Propagation Count** [33] that are often coined with user virality. Different from its definition in Chapter 3 where users only single infection with the a item, fan-out of user u, denoted by $V_{fo}(u)$, is now defined by the average number of times u propagates an item each time u adopts the item. Formally,

$$V_{fo}(u) = \frac{\sum_v \sum_x \mathbf{p}(u,x,v)}{\sum_x \mathbf{a}(u,x)} = \frac{\mathbf{p}(u,\cdot,\cdot)}{\mathbf{a}(u,\cdot)} \tag{4.1}$$

Table 4.1: Notations used to describe temporal and incremental models.

$\boldsymbol{\mathcal{U}}_t$/ $\boldsymbol{\mathcal{X}}_t$	Set of users/ content items up to time t
$\boldsymbol{\mathcal{U}}_t^{pro}$/ $\boldsymbol{\mathcal{U}}_t^{inf}$	Sets of propagating/ infected users up to time t
(u, x)	An adoption instance wherein user u adopts item x
(u, x, v)	A propagation instance wherein user u propagates item x to user v
$\boldsymbol{\mathcal{A}}_t(u, x)$	Bag of all (u, x) adoption instances up to time t
$\boldsymbol{\mathcal{P}}_t(u, x, v)$	Bag of all (u, x, v) propagation instances up to time t
$t(o)$	Time label of adoption/ propagation instance o
$\mathbf{a}(u, x)$	Number of times user u adopts item x
$\mathbf{a}_t(u, x)$	Temporally weighted variant of $\mathbf{a}(u, x)$ at time t
$\mathbf{p}(u, x, v)$	Number of times user u propagates item x to user v
$\mathbf{p}_t(u, x, v)$	Temporally weighted variant of $\mathbf{p}(u, x, v)$ at time t
$\overrightarrow{\mathbf{D}}^{pro}_{item}(u)$	Distribution of user u's propagation instances over all items that u propagates
$\overrightarrow{\mathbf{D}}^{pro}_{item,t}(u)$	Temporally weighted variant of $\overrightarrow{\mathbf{D}}^{pro}_{item}(u)$ at time t
$\overrightarrow{\mathbf{D}}^{pro}_{inf}(u)$	Distribution of user u's propagation instances over all users that u propagates item(s) to
$\overrightarrow{\mathbf{D}}^{pro}_{inf,t}(u)$	Temporally weighted variant of $\overrightarrow{\mathbf{D}}^{pro}_{inf}(u)$ at time t
$\overrightarrow{\mathbf{D}}^{item}_{pro}(x)$	Distribution of item x's propagation instances over all x' propagating users
$\overrightarrow{\mathbf{D}}^{item}_{pro,t}(x)$	Temporally weighted variant of $\overrightarrow{\mathbf{D}}^{item}_{pro}(x)$ at time t
$\overrightarrow{\mathbf{D}}^{item}_{inf}(x)$	Distribution of item x's propagation instances over all users infected with x
$\overrightarrow{\mathbf{D}}^{item}_{inf,t}(x)$	Temporally weighted variant of $\overrightarrow{\mathbf{D}}^{item}_{inf}(x)$ at time t
$\overrightarrow{\mathbf{D}}^{inf}_{item}(v)$	Distribution of user v's propagation instances over all items that v is infected with
$\overrightarrow{\mathbf{D}}^{inf}_{item,t}(v)$	Temporally weighted variant of $\overrightarrow{\mathbf{D}}^{inf}_{item}(v)$ at time t
$\overrightarrow{\mathbf{D}}^{inf}_{pro}(v)$	Distribution of user v's propagation instances over all users propagating item(s) to v
$\overrightarrow{\mathbf{D}}^{inf}_{pro,t}(v)$	Temporally weighted variant of $\overrightarrow{\mathbf{D}}^{inf}_{pro}(v)$ at time t
$E(\overrightarrow{\mathbf{D}})$	Entropy of distribution $\overrightarrow{\mathbf{D}}$
$H(\overrightarrow{\mathbf{D}})$	normalized entropy of distribution $\overrightarrow{\mathbf{D}}$
$V_{mm,t}(u)$	Virality of user u at time t as measured by model mm
$S_{mm,t}(v)$	Susceptibility of user v at time t as measured by model mm
$I_{mm,t}(x)$	Virality of item x at time t as measured by model mm

Propagation count of user u, denoted by $V_{pc}(u)$, is defined by the number of times u propagates item(s). That is,

$$V_{pc}(u) = \mathbf{p}(u, \cdot, \cdot) \tag{4.2}$$

Baseline for user susceptibility. In Chapter 3, we used Fan-In as baseline for user susceptibility, which is defined as the likelihood the user is infected with items when he is exposed to items. As mentioned in Section 4.1, the number of times a user is exposed to an item is usually not available. We therefore use **Infection Count** as baseline for user susceptibility. The infection count of user v, denoted by $S_{ic}(v)$, is the number of times v is infected with items. That is,

$$S_{ic}(v) = \sum_u \sum_x \mathbf{p}(u,x,v) = \mathbf{p}(\cdot,\cdot,v) \tag{4.3}$$

Baselines for content virality. Similar to Chapter 3, we again use **Popularity** and **Viral Coefficient** as baselines for content virality. In the settings that users may have multiple adoptions and infections with the same item, these measures are defined as follows.

Popularity of an item x, denoted by $I_p(x)$, can be measured by the number of times x is adopted, or the number of times x is propagated. However, highly adopted items are not always well propagated. Users may adopt them due to some external factors beyond what can be observed. Therefore, it is more reasonable to use the number of times an item is propagated to measure the item's popularity. That is,

$$I_p(x) = \sum_u \sum_v \mathbf{p}(u,x,v) = \mathbf{p}(\cdot,x,\cdot) \tag{4.4}$$

Viral coefficient of item x, denoted by $I_{vc}(x)$, is defined by the average numbers of propagation instances x gets per its adoption. That is,

$$I_{vc}(x) = \frac{\sum_u \sum_v \mathbf{p}(u,x,v)}{\sum_u \mathbf{a}(u,x)} = \frac{\mathbf{p}(\cdot,x,\cdot)}{\mathbf{a}(\cdot,x)} \tag{4.5}$$

4.3.3 Temporal Variants of Baseline Models

Given all the adoption and propagation instances up to time t, the static baseline models above only measure the cumulative user and item factors up to time t, but not the recent factors at time t. We now extend the baselines so that we can obtain the recent factors at t. In the following, we derive the counts or the measurements at time t by assigning temporal weights to the adoption and propagation instances giving us temporally weighted variants of $\mathbf{a}(u,x)$ and $\mathbf{p}(u,x,v)$ denoted by $\mathbf{a}_t(u,x)$ and $\mathbf{p}_t(u,x,v)$ respectively.

Temporal weighting scheme. A simple temporal weighting scheme is to let the adoption and propagation instances decay over time. That is, in counting the instances up to time t, each instance o with timestamp $t(o)$ is weighted by $\epsilon^{t-t(o)}$ where $\epsilon \in (0,1)$. However, this requires us to update $\mathbf{a}_t(u,x)$ and $\mathbf{p}_t(u,x,v)$ after each time step for, respectively, all pairs of u and x where u ever adopts x, and for all tuples of u, v, and x where u ever propagates x to v. This is computationally expensive. Furthermore, in each time step t, all the adoption (propagation) instances in t only belong to a small proportion of u and x pairs (u, v and x tuples). Therefore, instead of decaying weight of the old instances, we exponentially amplify weight of the new instances at each time step by $\frac{1}{\epsilon}$. For each time step, we thus only need to update $\mathbf{a}_t(u,x)$ if u adopts x at time t, and to update $\mathbf{p}_t(u,x,v)$ if u propagates x to v at time t. To increase the weight of the new instances, we assign to each instance o the weight $(1/\epsilon)^{t(o)}$. Obviously, this is relatively the same with time decaying weighting scheme. Now, $\mathbf{a}_t(u,x)$ and $\mathbf{p}_t(u,x,v)$ can incrementally updated:

$$\begin{aligned} \mathbf{a}_t(u,x) &= \sum_{o \in \boldsymbol{\mathcal{A}}_t(u,x)} (1/\epsilon)^{t(o)} \\ &= \mathbf{a}_{t-1}(u,x) + |\boldsymbol{\mathcal{A}}_t(u,x) - \boldsymbol{\mathcal{A}}_{t-1}(u,x)|(1/\epsilon)^t \end{aligned} \tag{4.6}$$

$$\begin{aligned} \mathbf{p}_t(u,x,v) &= \sum_{o \in \boldsymbol{\mathcal{P}}_t(u,x,v)} (1/\epsilon)^{t(o)} \\ &= \mathbf{p}_{t-1}(u,x,v) + |\boldsymbol{\mathcal{P}}_t(u,x,v) - \boldsymbol{\mathcal{P}}_{t-1}(u,x,v)|(1/\epsilon)^t \end{aligned} \tag{4.7}$$

where $|\boldsymbol{\mathcal{A}}_t(u,x) - \boldsymbol{\mathcal{A}}_{t-1}(u,x)|$ is simply the number of times u adopts x in time t. Similarly, $|\boldsymbol{\mathcal{P}}_t(u,x,v) - \boldsymbol{\mathcal{P}}_{t-1}(u,x,v)|$ is simply the number of times u propagates x to v in time t.

Temporal baseline models. With the temporally weighted adoption and propagation counts, the baseline models can now be extended to handle temporal propagation data by substituting the non-weighted counts by their temporally weighted ones. For example, the **Temporal Fan-out** of user u at time t, denoted by $V_{fo,t}(u)$, is defined as follows.

$$V_{fo,t}(u) = \frac{\mathbf{p}_t(u,\cdot,\cdot)}{\sum_x \mathbf{a}_t(u,x)} \tag{4.8}$$

In the similar spirit, we extend **Propagation Count**, **Infection Count**, **Popularity**, and **Viral Coefficient** to **Temporal Propagation Count**, **Temporal Infection Count**, **Temporal Popularity**, and **Temporal Viral Coefficient** respectively.

4.4 Mutual Dependency & Unbiased Models

We now develop new models that consider two principles: the inter-dependencies (or mutual dependency) between user and item factors (like in Chapter 3), and the unbiasness of each factor. There are also the static and temporal variants of the models as described below.

4.4.1 Model Principles

Our models are designed based on following principles.

Mutual dependencies among user virality, user susceptibility, and content virality, which we presented in Chapter 3, and are reminded below.

- A viral content item is one that can be propagated by non-viral users to non-susceptible users.
- A viral user is one who can propagate non-viral items to non-susceptible users.

- A susceptible user is one who can be infected with non-viral items that are propagated by non-viral users.

Unbiasness of user virality, user susceptibility, and content virality. This principle addresses the bias that may be introduced in measuring the user and content factors as we do not use user-item exposure. To do this, the principle requires the evidence supporting the factors would not be biased by a single user or item. That is,

- A viral item x should attract propagation instances across propagating users and across infected users. This prevents x from being biased by a single (or very few) user propagating or adopting x.

- A viral user should get propagation instances across items she propagates and across users she propagates items to. In other words, it not be the case that an (or very few) item or a (very few) user adopting u's items multiple times making u appears to be viral.

- A susceptible user should be easily infected across items and across users propagating items. This again prevents the susceptible user v from being biased by a single (or very few) item or a single (or very few) user infecting v with the item multiple times.

4.4.2 Static Mutual Dependency & Unbiased Model

We now present the static **Mutual Dependency & Unbiased Model** (**mdu** model), for measuring virality and susceptibility as follows.

- We let each propagation instance (u, x, v) represents a unit of propagation work.

- We assume that a user's virality is evenly distributed over all his propagation instances. A user's virality is then proportional to the number of propagation instances where the user plays the role of propagating user. Similarly, a user's susceptibility is proportional to the number of propagation instances where the user plays the role of infected user; and an item's virality is proportional to the number of propagation instances involving the item.

- To capture the mutual dependencies between user and item factors, we measure a factor by weighting each propagation instance with the inverse of other factors.

- A user's virality is proportional to the uniformity of distributions of her propagation instances over items she propagates, and over people she propagates the items to. Similarly, an items' virality is proportional to the uniformity of distributions of its propagation instances over users propagating the item, and over the users infected with the item. A user's susceptibility is proportional to the uniformity of distributions of her propagation instances over items she is infected with, and over the users propagating items to her. The above model feature is to be compliant with the unbiasness principle.

In **mdu** model, we use $V_{mdu}(u)$, $S_{mdu}(v)$, and $I_{mdu}(x)$ to denote the virality of user u, susceptibility of user v, and virality of item x respectively. We formalize the above assumptions into the following model equations:

$$V_{mdu}(u) = \left[\sum_{v}\sum_{x} \mathbf{p}(u,v,x)\left(1 - \frac{I_{mdu}(x)}{2\mathbf{p}(\cdot,\cdot,x)} - \frac{S_{mdu}(v)}{2\mathbf{p}(\cdot,v,\cdot)}\right)\right] \cdot \\ \cdot \left[H\big(\overrightarrow{\mathbf{D}}^{pro}_{item}(u)\big) \cdot H\big(\overrightarrow{\mathbf{D}}^{pro}_{inf}(u)\big)\right]^{\beta} \quad (4.9)$$

$$I_{mdu}(x) = \left[\sum_{u}\sum_{v} \mathbf{p}(u,x,v)\left(1 - \frac{V_{mdu}(u)}{2\mathbf{p}(u,\cdot,\cdot)} - \frac{S_{mdu}(v)}{2\mathbf{p}(\cdot,v,\cdot)}\right)\right] \cdot \\ \cdot \left[H\big(\overrightarrow{\mathbf{D}}^{item}_{pro}(x)\big) \cdot H\big(\overrightarrow{\mathbf{D}}^{item}_{inf}(x)\big)\right]^{\beta} \quad (4.10)$$

$$S_{mdu}(v) = \left[\sum_x \sum_u \mathbf{p}(u,v,x)\left(1 - \frac{I_{mdu}(x)}{2\mathbf{p}(\cdot,\cdot,x)} - \frac{V_{mdu}(u)}{2\mathbf{p}(u,\cdot,\cdot)}\right)\right]\cdot \\ \cdot\left[H\big(\overrightarrow{\mathbf{D}}^{inf}_{pro}(v)\big)\cdot H\big(\overrightarrow{\mathbf{D}}^{inf}_{item}(v)\big)\right]^{\beta} \quad (4.11)$$

In the right hand side of Equations 4.9, 4.10, and 4.11, the first part captures the mutual dependency among the user and item factors, and second part captures the unbiasness of the three factors. $\beta \geq 0$ is predefined parameter. We use β to moderate the weight of the unbiasness of the factors relative to their mutual dependency.

In Equation 4.9, $\overrightarrow{\mathbf{D}}^{pro}_{item}(u)$ is the distribution of user u's propagation instances over all items that u propagated. For example, if u only propagated x_1 and x_2, then we have $\overrightarrow{\mathbf{D}}^{pro}_{item}(u) = (\frac{\mathbf{p}(u,x_1,\cdot)}{\mathbf{p}(u,\cdot,\cdot)}, \frac{\mathbf{p}(u,x_2,\cdot)}{\mathbf{p}(u,\cdot,\cdot)})$. $\overrightarrow{\mathbf{D}}^{pro}_{inf}(u)$ is the distribution of user u's propagation instances over all users that u propagated item(s) to. For example, if u only propagated item(s) to v_1, v_2, and v_3, then $\overrightarrow{\mathbf{D}}^{pro}_{inf}(u) = (\frac{\mathbf{p}(u,\cdot,v_1)}{\mathbf{p}(u,\cdot,\cdot)}, \frac{\mathbf{p}(u,\cdot,v_2)}{\mathbf{p}(u,\cdot,\cdot)}, \frac{\mathbf{p}(u,\cdot,v_3)}{\mathbf{p}(u,\cdot,\cdot)})$. The unbiasness of virality of u then measured by entropies, $E(\overrightarrow{\mathbf{D}}^{pro}_{item}(u))$ and $E(\overrightarrow{\mathbf{D}}^{pro}_{inf}(u))$.

Similarly, we have $\overrightarrow{\mathbf{D}}^{item}_{pro}(x)$ and $\overrightarrow{\mathbf{D}}^{item}_{inf}(x)$ are the distributions of item x's propagation instances over all x' propagating users and over all users infected with x respectively. These distributions' entropies are used to measure the unbiasness of virality of item x in Equation 4.10. In Equation 4.11, $\overrightarrow{\mathbf{D}}^{inf}_{item}(v)$ and $\overrightarrow{\mathbf{D}}^{inf}_{pro}(v)$ are the distributions of user v's propagation instances over all items that v is infected with and over all users propagating item(s) to v respectively, and these distributions' entropies are used to measure the unbiasness of susceptibility of user v.

Since the entropies of different distributions generally have different scales. We use the *normalized entropies* in the right hand sides of Equations 4.9, 4.10, and 4.11. For a distribution $\overrightarrow{\mathbf{D}}$, $H(\overrightarrow{\mathbf{D}})$ is normalized entropy of $\overrightarrow{\mathbf{D}}$ which measures the uniformity of $\overrightarrow{\mathbf{D}}$ is. $H(\overrightarrow{\mathbf{D}})$ is defined as follows.

$$H(\overrightarrow{\mathbf{D}}) = \delta + (1-\delta)\frac{E(\overrightarrow{\mathbf{D}})}{\ln dim(\overrightarrow{\mathbf{D}})} \tag{4.12}$$

where $\delta \in (0,1)$; $dim(\overrightarrow{\mathbf{D}})$ is the dimension of $\overrightarrow{\mathbf{D}}$; and $E(\overrightarrow{\mathbf{D}})$ is the entropy of $\overrightarrow{\mathbf{D}}$. Obviously we have, $E(\overrightarrow{\mathbf{D}}) \in [0, \ln dim(\overrightarrow{\mathbf{D}})]$, making $H(\overrightarrow{\mathbf{D}}) \in [\delta, 1]$.

$H(\overrightarrow{\mathbf{D}}^{pro}_{item}(u))$ and $H(\overrightarrow{\mathbf{D}}^{pro}_{inf}(u))$ equals to δ when u propagated only one item and when u propagated item(s) to only one user respectively. Similarly $H(\overrightarrow{\mathbf{D}}^{item}_{pro}(x))$ and $H(\overrightarrow{\mathbf{D}}^{item}_{inf}(x))$ equals to δ when x is propagated by only one user and when x is infected with by only one user respectively. $H(\overrightarrow{\mathbf{D}}^{inf}_{item}(v))$ and $H(\overrightarrow{\mathbf{D}}^{inf}_{pro}(v))$ equals to δ when v is infected with only one item and when there is only one user propagates item(s) to v. Moreover, it is expected that a user (an item) ever propagated (be propagated) has some degree of virality, and that a user ever be infected with item(s) to have some degree of susceptibility. We therefore set δ to a positive value.

Special cases. In the case $\beta = 0$, the unique solution for users' virality and susceptibility and items's virality is $\frac{\mathbf{p}(u,\cdot,\cdot)}{2}$, $\frac{\mathbf{p}(\cdot,v,\cdot)}{2}$ and $\frac{\mathbf{p}(\cdot,\cdot,x)}{2}$ for all users u, v and all items x respectively. In other words, when $\beta = 0$, the **mdu** model degenerates to **Propagation Count**, **Infection Count**, and **Popularity** baselines for user virality and susceptibility and item virality respectively. Similarly, when $\beta >> 1$, the **mdu** model shares some similarity with **Fan-out** and **Viral Coefficient** baselines for user and item virality.

4.4.3 Temporal Mutual Dependency & Unbiased Model

We now extend the **mdu** model further to a temporal model, called the **Temporal Mutual Dependency & Unbiased Model** (**t-mdu** model). Similar to extending the static baselines, the main idea here is to use temporally weighted variants of number of propagation instances, and those of distributions of propagation instances over users and items. That is, at time t, the temporal susceptibility $S_{mdu,t}(v)$ of user v is computed from users' temporal virality $V_{mdu,t}(u)$ and items' temporal virality $I_{mdu,t}(x)$ as follows.

$$S_{mdu,t}(v) = \left[\sum_{x}\sum_{u} \mathbf{p}_t(u,v,x)\left(1 - \frac{I_{mdu,t}(x)}{2\mathbf{p}_t(\cdot,\cdot,x)} - \frac{V_{mdu,t}(u)}{2\mathbf{p}_t(u,\cdot,\cdot)}\right)\right] \cdot \cdot \left[H\left(\overrightarrow{\mathbf{D}}^{inf}_{pro,t}(v)\right) \cdot H\left(\overrightarrow{\mathbf{D}}^{inf}_{item,t}(v)\right)\right]^{\beta} \quad (4.13)$$

where $\overrightarrow{\mathbf{D}}^{inf}_{pro,t}(v)$ and $\overrightarrow{\mathbf{D}}^{inf}_{item,t}(v)$ is the temporally weighted versions of $\overrightarrow{\mathbf{D}}^{inf}_{pro}(v)$ and $\overrightarrow{\mathbf{D}}^{inf}_{item}(v)$ at time t respectively. For example, if up to time t, only u_1 and u_2 propagated item(s) to v then we have $\overrightarrow{\mathbf{D}}^{inf}_{pro,t}(v) = (\frac{\mathbf{p}_t(u_1,v,\cdot)}{\mathbf{p}(\cdot,v,\cdot)}, \frac{\mathbf{p}_t(u_2,v,\cdot)}{\mathbf{p}(\cdot,v,\cdot)})$. If up to time t, v is only infected with items x_1, x_2, and x_3 then we have $\overrightarrow{\mathbf{D}}^{inf}_{item,t}(v) = (\frac{\mathbf{p}_t(\cdot,v,x_1)}{\mathbf{p}_t(\cdot,v,\cdot)}, \frac{\mathbf{p}_t(\cdot,v,x_2)}{\mathbf{p}_t(\cdot,v,\cdot)}, \frac{\mathbf{p}_t(\cdot,v,x_3)}{\mathbf{p}_t(\cdot,v,\cdot)})$.

Similarly, user virality and susceptibility of u are defined as follows.

$$V_{mdu,t}(u) = \left[\sum_{v}\sum_{x} \mathbf{p}_t(u,v,x)\left(1 - \frac{I_{mdu,t}(x)}{2\mathbf{p}_t(\cdot,\cdot,x)} - \frac{S_{mdu,t}(v)}{2\mathbf{p}_t(\cdot,v,\cdot)}\right)\right] \cdot \cdot \left[H\left(\overrightarrow{\mathbf{D}}^{pro}_{item,t}(u)\right) \cdot H\left(\overrightarrow{\mathbf{D}}^{pro}_{inf,t}(u)\right)\right]^{\beta} \quad (4.14)$$

$$S_{mdu,t}(v) = \left[\sum_{x}\sum_{u} \mathbf{p}_t(u,v,x)\left(1 - \frac{I_{mdu,t}(x)}{2\mathbf{p}_t(\cdot,\cdot,x)} - \frac{V_{mdu,t}(u)}{2\mathbf{p}_t(u,\cdot,\cdot)}\right)\right] \cdot \cdot \left[H\left(\overrightarrow{\mathbf{D}}^{inf}_{pro,t}(v)\right) \cdot H\left(\overrightarrow{\mathbf{D}}^{inf}_{item,t}(v)\right)\right]^{\beta} \quad (4.15)$$

where $\overrightarrow{\mathbf{D}}^{pro}_{item,t}(v)$, $\overrightarrow{\mathbf{D}}^{prof}_{inf,t}(v)$, $\overrightarrow{\mathbf{D}}^{inf}_{item,t}(v)$, and $\overrightarrow{\mathbf{D}}^{inf}_{pro,t}(v)$ is the temporally weighted versions of $\overrightarrow{\mathbf{D}}^{pro}_{item}(v)$, $\overrightarrow{\mathbf{D}}^{pro}_{inf}(v)$, $\overrightarrow{\mathbf{D}}^{inf}_{item}(v)$ and $\overrightarrow{\mathbf{D}}^{inf}_{pro}(v)$ at time t respectively.

4.4.4 Model Learning

We employ the following **iterative computation method** for computing users' and items' factors in the **mdu** and **t-mdu** models.

For the **mdu** model, starting from a random initialization of these factors, we iteratively update each of these factors using the others based on their inter-relationship as defined in Equations 4.9, 4.10, and 4.11. This process repeats until we reach a predefined maximum number of iterations or when the values converge. Similarly, we can compute the scores in **t-mdu** model using the same iterative method and the corresponding updating equations of the model.

It can be shown that the right hand sides of Equation 4.9, 4.10, and 4.11 form a contraction map [253] in the following subspace

$$\boldsymbol{\mathcal{X}} = \prod_{x \in \boldsymbol{\mathcal{X}}} [0, \mathbf{p}(\cdot, x, \cdot)] \times \prod_{u \in \boldsymbol{\mathcal{U}}^{pro}} [0, \mathbf{p}(u, \cdot, \cdot)] \times \prod_{v \in \boldsymbol{\mathcal{U}}^{inf}} [0, \mathbf{p}(\cdot, \cdot, v)]$$

where $\prod$ is the Cartesian product of sets. This means if we initialize $I_{mdu}(x)$, $V_{mdu}(u)$, and $S_{mdu}(v)$ by any random value in $[0, \mathbf{p}(\cdot, x, \cdot)]$, $[0, \mathbf{p}(u, \cdot, \cdot)]$, and $[0, \mathbf{p}(\cdot, \cdot, v)]$ respectively, the above iterative computation method for the **mdu** model always converges to a unique solution, independent of the initialized values. Similarly, we can prove that the iterative computation method for **t-mdu** model converges to a unique solution.

Implementation notes. We empirically found that the following initialization of the factors results in the iterative computation method converges much faster.

- Initializing user u's virality by the **Propagation Count** or **Temporal Propagation Count** models, i.e., $V_{pc}(u)$ or $V_{pc,t}(u)$
- Initializing user v's susceptibility by the **Infection Count** or **Temporal Infection Count** models, i.e., $S_{ic}(v)$ or $S_{ic,t}(v)$
- Initializing item x's virality by the **Popularity** or **Temporal Popularity** models, i.e., $I_p(x)$ or $I_{p,t}(x)$

4.5 Incremental Model

In both **mdu** and **t-mdu** models, the iterative computation method used for learning the factors may require many iterations incurring significant computation overheads. We therefore introduce in this section an incremental model for working with data streams.

4.5.1 Overview of the Incremental Approach

The main idea of incremental method is to first find an assignment of the whole amount propagation work to users (without involving the items). We are interested in the assignment that can be incrementally updated. We then correct this biased assignment by using only one iteration of the iterative computation method. At each time step t, our incremental method includes the following steps.

Step 1: Incremental updating propagation rank of users. In this step, we assign the whole amount of propagation work carried by all propagation instances to the users only. We assign to each user $u \in \boldsymbol{\mathcal{U}}_t^{pro}$ a **Propagation Rank** score $\pi_t^{pro}(u)$ based on propagation instances wherein the user propagated item(s) to other users. Similarly, we assign to each user $v \in \boldsymbol{\mathcal{U}}_t^{inf}$ a **Propagation Rank** score $\pi_t^{inf}(v)$ based on propagation instances wherein the user is infected with item(s). We use $\pi_t^{pro}(u)$ to denote propagation rank of u when u plays the role of propagating user, and use $\pi_t^{inf}(v)$ to denote propagation rank of v when v plays the role of infected user. We aim to find the ranks that are unit normalized, i.e., $\sum_{u \in \boldsymbol{\mathcal{U}}_t^{pro}} \pi_t^{pro}(u) + \sum_{v \in \boldsymbol{\mathcal{U}}_t^{inf}} \pi_t^{inf}(v) = 1$, and can be updated incrementally. For a clear presentation, we will describe the rank's definition and its incremental update in Section 4.5.3.

Step 2: Computing users' virality and susceptibility and items' virality using the Propagation Rank. In this step, our approach to compute user and item factors using propagation rank is motivated by the following principles, which are adaptations of the mutual dependency principle (see Section 4.4).

- A viral content item is an item that can be propagated from low propagation rank propagating users to low propagation rank infected users.
- A viral user is a user who propagates less viral items to low propagation rank infected users.
- A susceptible user is a user who can be infected with less viral items that are propagated to her by less viral users.

Once we have obtained the Propagation Rank of the users, we compute item virality similar to the **mdu** and **t-mdu** models. That is,

$$I_{inc,t}(x) = \left[\sum_u \sum_v \mathbf{p}_t(u,x,v) \left(1 - \pi_t^{pro}(u) - \pi_t^{inf}(v) \right) \right] \cdot \\ \cdot \left[H\left(\overrightarrow{\mathbf{D}}^{item}_{pro,t}(x)\right) \cdot H\left(\overrightarrow{\mathbf{D}}^{item}_{inf,t}(x)\right) \right]^{\beta} \tag{4.16}$$

This is followed by user virality as follows.

$$V_{inc,t}(u) = \left[\sum_x \sum_x \mathbf{p}_t(u,v,x) \left(1 - \frac{I_{inc,t}(x)}{2\mathbf{p}(\cdot,\cdot,x)} - \pi_t^{inf}(v) \right) \right] \cdot \\ \cdot \left[H\left(\overrightarrow{\mathbf{D}}^{pro}_{item,t}(u)\right) \cdot H\left(\overrightarrow{\mathbf{D}}^{pro}_{inf,t}(u)\right) \right]^{\beta} \tag{4.17}$$

Finally, we can compute user susceptibility scores using the following equation, which is similar to Equation 4.15.

$$S_{inc,t}(v) = \left[\sum_x \sum_u \mathbf{p}_t(u,v,x) \left(1 - \frac{I_{inc,t}(x)}{2\mathbf{p}_t(\cdot,\cdot,x)} - \frac{V_{inc,t}(u)}{2\mathbf{p}_t(u,\cdot,\cdot)} \right) \right] \cdot \\ \cdot \left[H\left(\overrightarrow{\mathbf{D}}^{inf}_{pro,t}(v)\right) \cdot H\left(\overrightarrow{\mathbf{D}}^{inf}_{item,t}(v)\right) \right]^{\beta} \tag{4.18}$$

4.5.2 Propagation Rank

We now define the propagation rank of a user. The propagation rank is essentially the Pagerank of users in a specially constructed *propagation graph* as follows.

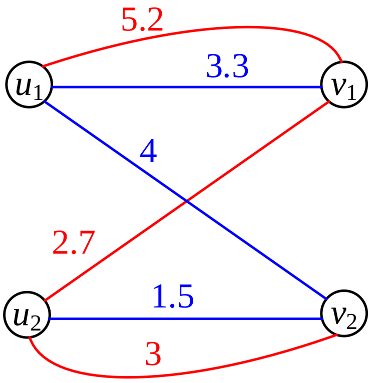
Propagation graph. At a time t, the propagation graph G_t^d is a weighted bipartite multigraph where the nodes are $\boldsymbol{\mathcal{U}}_t^{pro} \cup \boldsymbol{\mathcal{U}}_t^{inf}$. As there may be users belonging to both $\boldsymbol{\mathcal{U}}_t^{pro}$ and $\boldsymbol{\mathcal{U}}_t^{inf}$, a user may have two corresponding nodes in G_t^d, one for propagating user role and another for infected user role. For each user pair (u,v), $u \in \boldsymbol{\mathcal{U}}_t^{pro}$ and $v \in \boldsymbol{\mathcal{U}}_t^{inf}$, if u propagated item x to v, then we have in G_t^d an edge $\mathbf{e}_{u,x,v}$ that joins u and v. The weight of this edge is $\mathbf{p}_t(u,x,v)$. Note that there may more than one edge between a pair of users u and v as u may propagate many items to v.

Figure 4.2 shows an example of a propagation graph constructed from the provided propagation logs. In this example, u_1 propagated both x_1 and x_2 to v_1. Hence, in the constructed propagation graph, there are two edges joining u_1 and v_1. These edges have weight $\mathbf{p}_t(u_1,x_1,v_1)$ and $\mathbf{p}_t(u_1,x_2,v_1)$ respectively. The remaining edges are weighted similarly.

Ranking on propagation graph. The propagation graph G_t^d represents the relationship between the users propagating items and the users infected with items, regardless of the items. We therefore associate each edge from u to v (or vice versa) with an amount of propagation work that u assigns to v (or v assigns to u) to complete. The PageRank vector of G_t^d hence can be considered as the distribution of total amount of propagation work (measured by the propagation instances that are used to construct G_t^d) over all the nodes in G_t^d. In other words, the pagerank score of every node in G_t^d is its propagation rank as defined below.

u	x	v	$\mathbf{p}_t(u, x, v)$
u_1	x_1	v_1	5.2
u_1	x_2	v_1	3.3
u_1	x_2	v_2	4
u_2	x_1	v_1	2.7
u_2	x_1	v_2	3
u_2	x_2	v_2	1.5

(a)



(b)

Figure 4.2: Illustrative example of (a) propagation logs, and (b) the propagation graph constructed from the logs.

$$\begin{aligned}\pi_t^{pro}(u) &= \frac{c}{|\mathcal{U}_t^{pro}| + |\mathcal{U}_t^{pro}|} + (1-c)\sum_{v \in \mathcal{U}_t^{inf}} \frac{\mathbf{p}_t(u,x,v)}{\mathbf{p}_t(u,\cdot,\cdot)}\pi_t^{inf}(v) \\ \pi_t^{inf}(v) &= \frac{c}{|\mathcal{U}_t^{pro}| + |\mathcal{U}_t^{pro}|} + (1-c)\sum_{u \in \mathcal{U}_t^{pro}} \frac{\mathbf{p}_t(u,x,v)}{\mathbf{p}_t(\cdot,\cdot,v)}\pi_t^{pro}(u)\end{aligned}$$

where $c \in (0, 1)$ is the "damping factor", which is typically set to 0.15.

4.5.3 Incremental Computation of Propagation Rank

Approximating propagation rank by random walks. To compute the propagation rank scores incrementally, we employ the Monte Carlo methods [13] by conducting random walks on G_t^d as follows. Each random walk starts from a node of G_t^d for a maximum length of L for a small L (e.g., 20). We therefore have $|\mathcal{U}_t^{pro}| + |\mathcal{U}_t^{inf}|$ random walks. At each step of the random walk, we stop the walk with probability c, and otherwise traverse an edge randomly selected from all the edges of the current node. The probability that an edge is chosen is proportional to the weight of the edge. For each node n in G_t^d, we use

$\boldsymbol{\mathcal{V}}_t(n)$ to denote the number of times n is visited by all the random walks, and let $\boldsymbol{\mathcal{V}}_t(\cdot) = \sum_{n' \in \boldsymbol{\mathcal{U}}^{pro} \cup \boldsymbol{\mathcal{U}}^{inf}} \boldsymbol{\mathcal{V}}_t(n')$. Then, it is proven in [13, 14] that $\frac{\boldsymbol{\mathcal{V}}_t(n)}{\boldsymbol{\mathcal{V}}_t(\cdot)}$ heavily concentrates around its expectation which is the pagerank score of node n. In other words, for each $u \in \boldsymbol{\mathcal{U}}_t^{pro}$, $\pi_t^{pro}(u)$ is very well approximated by $\frac{\boldsymbol{\mathcal{V}}_t(u)}{\boldsymbol{\mathcal{V}}_t(\cdot)}$. Similarly, for each $v \in \boldsymbol{\mathcal{U}}_t^{inf}$, $\pi_t^{inf}(v)$ is very well approximated by $\frac{\boldsymbol{\mathcal{V}}_t(v)}{\boldsymbol{\mathcal{V}}_t(\cdot)}$.

Sampling random walks in amortized linear time. In unweighted graphs as studied in the previous works [63, 13, 14], the time complexity for sampling an edge in each step of the random walks is constant. However, in our case, since G_t^d is weighted graph, a naive method for sampling the edge will have the complexity of $\boldsymbol{\mathcal{O}}(m)$ where m is the number of edges of the current node of the step. This complexity is very high as we have to perform the sampling at each node of G_t^d for a large number of times. Furthermore, the larger m is, the more times we will have to perform the sampling at the node as it is visited by many more walks. We therefore make use of the **alias sampling method** [220] which allows to sample edges in amortized constant time. In this method, we first build for each node in G_t^d an "alias" of the node's edges. Then, based on this alias, an edge can be sampled in constant time, yet satisfying the probability that an edge is sampled is proportional to the edge's weight. The cost for building an alias is $\boldsymbol{\mathcal{O}}(m)$, making the cost for sampling (many) edges of the current node is amortized constant. This leads to the over all cost for sampling the random walks is amortized linear.

Incremental estimation of propagation rank. In our temporal weighting scheme (see Section 4.3.3), when given a new propagation instance (u, x, v), only the weights of edges of u and v may change, while other edges in G_t^d remain unchanged. Hence, only random walks that visit u or v are affected by the new propagation instance. We first remove all the random walks that visit u or v. We then re-conduct those walks from their start nodes, and update the visit counts and the approximation of propagation rank accordingly.

4.5.4 Cost Analysis

We now examine the cost of updating the approximation for the propagation rank vector presented above. We use $W_t(u)$, $W_t(v)$, and $W_t(u\|v)$ to denote the number of random walks on G_t^d that visit u, v, and visit u or v respectively. We then have to re-conduct $W_t(u\|v)$ walks that visits u or v to update the approximation. Obviously, we have

$$W_t(u\|v) \leq W_t(u) + W_t(v) \leq \boldsymbol{\mathcal{V}}_t(u) + \boldsymbol{\mathcal{V}}_t(v) = (\pi_t^{pro}(u) + \pi_t^{inf}(v))\boldsymbol{\mathcal{V}}_t(\cdot) \quad (4.19)$$

Due to the monotonicity of expectation, we have

$$\begin{aligned} \mathbf{E}\big[W(u\|v)\big] &\leq \mathbf{E}\big[(\pi_t^{pro}(u) + \pi_t^{inf}(v))\boldsymbol{\mathcal{V}}(.)\big] = \\ &= \mathbf{E}\big[\pi_t^{pro}(u) + \pi_t^{inf}(v)\big] \cdot \mathbf{E}\big[\boldsymbol{\mathcal{V}}(\cdot)\big] \end{aligned} \quad (4.20)$$

As shown in [136, 14], in a power law graph, pagerank score and degree of nodes follow the power law distributions with the same exponent $\theta \in (0, 1)$. That is, if π_t^j is the j^{th} largest entry of propagation rank vector π_t, then we have

$$\pi_t^j \sim \mathbf{C} j^{-\theta} \quad (4.21)$$

where $\mathbf{C}$ is the normalization constant such that $\sum_{j=1}^{N} \pi_t^j = 1$ where N is the number of nodes in graph, $N = |\boldsymbol{\mathcal{U}}_t^{pro} + \boldsymbol{\mathcal{U}}_t^{inf}|$. Therefore,

$$1 = \sum_{j=1}^{N} Cj^{-\theta} = CN^{1-\theta} \sum_{j=1}^{N} \big[\frac{1}{N}(\frac{j}{N})^{-\theta}\big] \approx CN^{1-\theta} \int_0^1 x^{-\theta} dx = \mathbf{C}\frac{N^{1-\theta}}{1-\theta}$$

Hence, $\mathbf{C} \approx \frac{1-\theta}{N^{1-\theta}}$, and therefore

$$\mathbf{E}\big[\pi_t(u) + \pi_t(v)\big] \leq \mathbf{E}\big[\pi_t(u)\big] + \mathbf{E}\big[\pi_t(v)\big] \leq 2 \cdot \pi_t^1 \approx 2\frac{1-\theta}{N^{1-\theta}} \quad (4.22)$$

Also, since at each random step, we stop the walk with probability c, the length $\mathbf{l}$ of each random walk has expectation $\mathbf{E}\big[\mathbf{l}\big] = 1/c$. Then, we have

$$\mathbf{E}\big[\boldsymbol{\mathcal{V}}(\cdot)\big] = N\mathbf{E}\big[\mathbf{l}\big] = \frac{N}{\epsilon} \tag{4.23}$$

From Equations 4.20, 4.22, and 4.23, we have

$$\mathbf{E}\big[W(u\|v)\big] \approx 2\frac{1-\theta}{N^{1-\theta}} \cdot \frac{N}{c} = 2\frac{1-\theta}{cN^{-\theta}} \tag{4.24}$$

This means that in each incremental update, we only have to re-conduct a small proportion of existing random walks.

Implementation notes. In practice, each instance propagation does not significantly change the user and item factors. Moreover, the propagation instances of viral or susceptible user or of viral items often occur within a short time window, leading to a large overlap between the sets of walks to be re-conducted. It is therefore more practical not to perform the updating for every new propagation instance, but after accumulating them for some short time window.

4.6 Experimental Evaluation

In this section, we evaluate the proposed models in three experiments. Firstly, we compare the user and item factors obtained by the different models. We examine some case examples to better illustrate the differences between the models. Secondly, we evaluate the accuracy of the models. Finally, we evaluate the incremental model by speedup and accuracy.

4.6.1 Dataset

Data collection. The dataset used in this work was collected from Twitter by a snowball sampling based crawler. We first manually selected a set of highly followed Twitter users in Singapore. They include the accounts of local sport and entertainment celebrities, political parties, politicians, mass media and bloggers, etc.. We expanded this set of users by adding more Singapore-based users[1] that are at most two hops away from some user in the original set. Using Twitter Stream APIs[2], we then obtained all tweets and retweets by the users in the set. In this work, we use all tweets in October 2014 to simulate a live tweet stream. This set includes 35,491,260 tweets and retweets posted by 525,632 users.

Item adoption and propagation. Like in Chapter 3, we again use hashtag as an item. We consider a user u adopts a hashtag when u posts an original tweet containing the hashtag. Also, if user v retweets an original tweets from u that contains a hashtag h, u is said to propagate h to v. We filtered away hashtags shorter than 2 characters excluding the # symbol. These short hashtags do not have clear semantics and are often the prefix of other truncated hashtags due to 140 characters length constraint. We also excluded hashtags longer than 20 characters as such hashtags are unpopular.

Figure 4.3 (a) shows that the dataset is very large with more than 140K users propagating at least one item, more than 400K users infected with some item, and 120K items being propagated. Figure 4.3 (b) shows the distributions of number of distribution instances over users and items in log-log scale. The figure shows that the dataset has power law-like distributions with most users propagated (or infected with) only a few items, and most of items are propagated (infected) by only a few users.

4.6.2 Experiment Settings

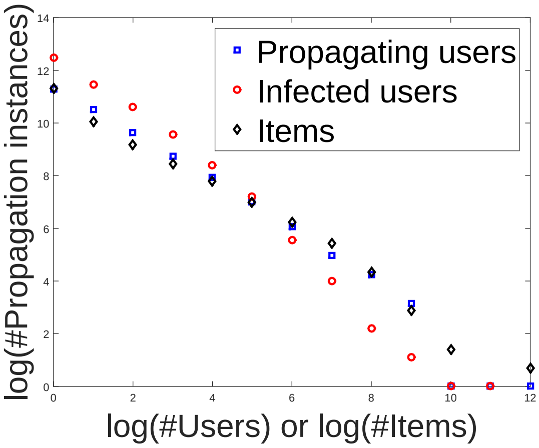
In our experiments, we set each time step to one hour, or 743 time steps in the month of October 2014. For temporal models, the temporal weight $\frac{1}{\epsilon}$ is set to $2^{1/24}$, implying that each adoption/ propagation instance decays by a half after one day, or almost completely decayed after 5 days (or 120 hours) as suggested in [236]. For both static and temporal mutual dependency

[1] A Twitter user is considered Singapore-based user if her profile location is Singapore

[2] https://dev.twitter.com/streaming/overview

#propagating users	143,169
#infected users	419,428
#items	123,542
#propagation instances	2,824,494
#time steps	743

(a)



(b)

Figure 4.3: (a) Statistics of the experimental dataset, and (b) Distributions of propagation instances over users and items

& unbiased models, the weight β is empirically set to 2. In approximating propagation rank vector by random walks, we use parameters that are used in previous works [63, 13, 14]. That is, to set “damping factor” c to 0.15, the maximum length of each random walk L to 20. Lastly, we set the time window for updating the approximation to 15 seconds.

For convenience, we use summary in Table 4.2 the acronyms used to denote different models for user virality and susceptibility and item virality.

4.6.3 Score Analysis

Similarity between models. We now compare the models in measuring user and item factors. For each model and at each time step, we compute Pearson Correlation coefficient between the score ranks of a pair of models. Tables 4.3 (a), (b), and (c) show the average the Pearson Correlation Coefficients (PCC) between different models across all time steps of the dataset. We arrive at the following findings. First, as expected, the models of the same type, i.e., static

Table 4.2: Acronyms for different models for user virality and susceptibility and item virality

Acronym	Model	Target factor
pc	Static Propagation Count	User virality
t-pc	Temporal Propagation Count	User virality
fo	Static Fan-Out	User virality
t-fo	Temporal Fan-Out	User virality
ic	Static Infection Count	User susceptibility
t-ic	Temporal Infection Count	User susceptibility
p	Static Popularity	Item virality
t-p	Temporal Popularity	Item virality
vc	Static Viral Coefficient	Item virality
t-vc	Temporal Viral Coefficient	Item virality
mdu	Static Mutual Dependency & Unbiased Model	All three factors
t-mdu	Temporal Mutual Dependency & Unbiased Model	All three factors
inc	Incremental Model	All three factors

or temporal, are quite correlated one with each other (with PCC ≤ 0.50) but not always correlated with models of other types. Second, count based baselines (**pc**, **t-pc**, **p** and **t-p**) and proportion based baselines (**fo**, **t-fo**, **vc** and **t-vc**) are not highly correlated as the correlation coefficients between them are not high. Thirdly, the **mdu** and **t-mdu** are similar to but not the same with count based baselines **pc** and **t-pc**. This is expected as count based models are special cases of the mutual dependency & consistency models. Lastly, the **t-mdu** model and incremental model are very similar in ranking users and items by their virality (susceptibility).

Temporal-MDU vs Incremental Model: top users and items. We further evaluate the similarity between **t-mdu** and **inc** models by comparing their top K viral users, susceptible users, and viral items. Figure 4.4 shows the average Jaccard coefficient of top K users/items over all the time steps with K varied from 5 to 100. The figure clearly shows that the tops K's of the two models are consistently highly similar (≥ 0.9) across different K's. This means **t-mdu** and **inc** models produce almost the same top viral (susceptible) users and items.

Dynamics of user and item factors. We first study the dynamics of user/item factors by examining the changes to the top 10 users/items returned by different models. We use Jaccard coefficients to measure the difference of

Table 4.3: Average Pearson rank correlation coefficients between scores for (a) users' virality and (b) susceptibility, and (c) items' virality obtained by different methods.

(a)

	pc	**t-pc**	**fo**	**t-fo**	**mdu**	**t-mdu**	**inc**
pc	1	0.50	0.55	0.43	**0.91**	0.60	0.60
t-pc	-	1	0.18	0.19	0.50	**0.97**	**0.97**
fo	-	-	1	**0.82**	0.46	0.24	0.24
t-fo	-	-	-	1	0.40	0.25	0.25
mdu	-	-	-	-	1	0.63	0.62
t-mdu	-	-	-	-	-	1	**0.99**

(b)

	ic	**t-ic**	**mdu**	**t-mdu**	**inc**
ic	1	0.49	**0.82**	0.56	0.56
t-ic	-	1	0.46	**0.97**	**0.97**
mdu	-	-	1	0.58	0.58
t-mdu	-	-	-	1	**0.99**

(c)

	p	**t-p**	**vc**	**t-vc**	**mdu**	**t-mdu**	**inc**
p	1	**0.83**	0.26	0.20	**0.91**	**0.86**	**0.87**
t-p	-	1	0.41	0.41	**0.80**	**0.93**	**0.95**
vc	-	-	1	**0.93**	0.14	0.30	0.32
t-vc	-	-	-	1	0.13	0.30	0.31
mdu	-	-	-	-	1	**0.90**	**0.87**
t-mdu	-	-	-	-	-	1	**0.88**

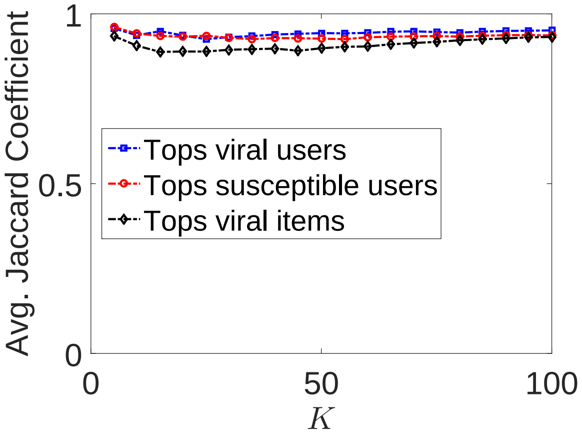
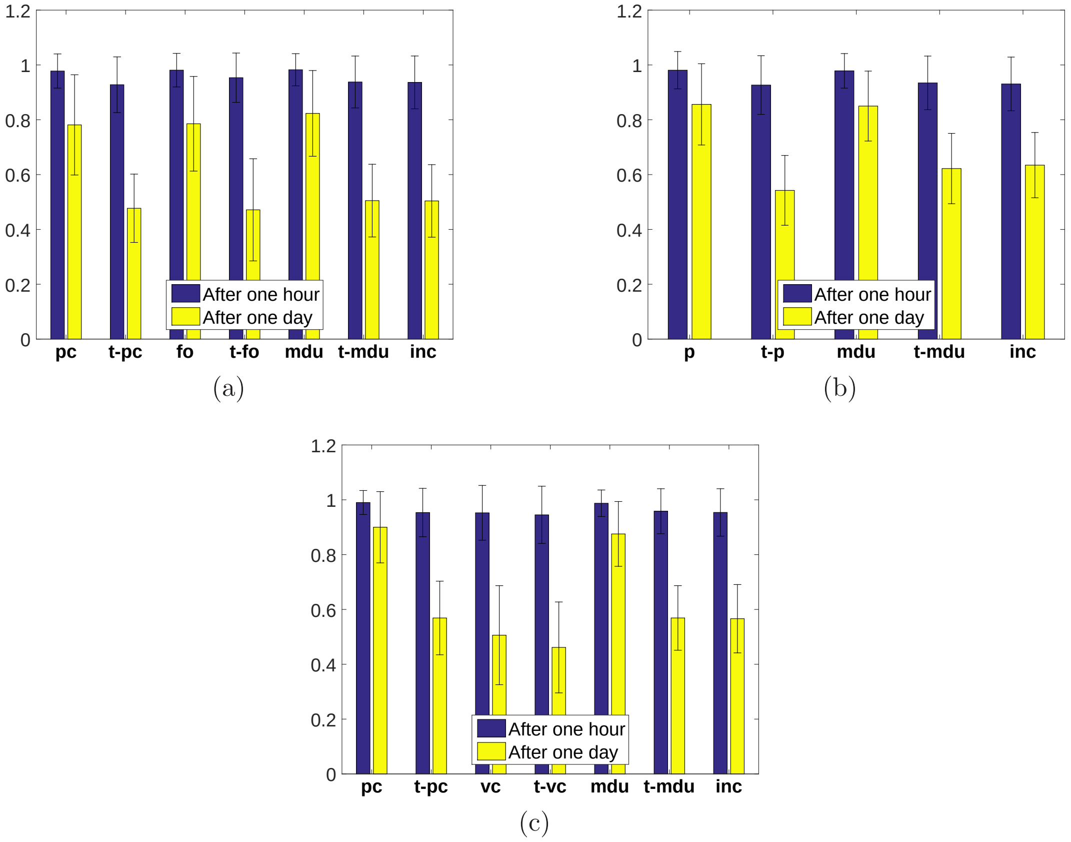


Figure 4.4: Average Jaccard coefficient between top K users and items returned by the **t-mdu** and **inc** models across time steps.

top 10 users/items between the initial ranks and ranks obtained after one hour and after one day. Figures 4.5 (a), (b), and (c) show that small changes to top 10 users/items occur after one hour, but the changes are more drastic after one day. This suggests that it is important to create temporal models to track



(a) (b) (c)

Figure 4.5: Dynamicity of different models as measured by the average Jaccard coefficients between (a) top viral users, (b) top susceptible users, and (c) tops viral items after one hour and one day.

the recent factor values.

4.6.4 Case Examples

We now show the differences among the models using some examples of their ranking results.

Viral user example. Figure 4.4 (a) shows the profiles of two users having very different number of propagation instances. User-a is a entertainment celebrity, and user-b is a sport fan club. user-b has the number of propagation instances (**pc**) and fan-out (**fo**) 5 and 10 times respectively higher than those of user-a. However, users-a propagated items more diversely (as measured by $H(\overrightarrow{\mathbf{D}}^{pro}_{item})$). Hence, in the **mdu** model, user-b's virality score is only 4 times higher than user-a. Moreover, in the current time step ($t = 360$), user-a

Table 4.4: Case examples of (a) viral users, (b) susceptible users, and (c) viral items. **#ni** is the number of new propagation instance in the time step.

(a) Profile of example viral users at time t = 360

	dc	**t-dc**	**fo**	**t-fo**	$H(\overrightarrow{\mathbf{D}}^{pro}_{item})$	$H(\overrightarrow{\mathbf{D}}^{pro}_{item,t})$	$H(\overrightarrow{\mathbf{D}}^{pro}_{inf})$	$H(\overrightarrow{\mathbf{D}}^{pro}_{inf,t})$	**#ni**	**mdu**	**inc**	**t-mdu**
user-a	31,120	2,786.0	42.9	34.6	0.87	0.86	0.96	0.92	227	18,772.2	1,568.6	1,585.4
user-b	157,320	5,043.9	561.9	403.7	0.77	0.59	0.96	0.89	124	73,175.0	1,228.7	1,256.8

(b) Profile of example susceptible users at time t = 360

	ic	**t-ic**	$H(\overrightarrow{\mathbf{D}}^{pro}_{item})$	$H(\overrightarrow{\mathbf{D}}^{pro}_{item,t})$	$H(\overrightarrow{\mathbf{D}}^{pro}_{inf})$	$H(\overrightarrow{\mathbf{D}}^{pro}_{inf,t})$	**#ni**	**mdu**	**inc**	**t-mdu**
user-c	398	6,224,299.4	0.7	0.7	0.9	0.9	11	119.5	52.3	61.9
user-d	878	4,006,310.0	0.8	0.8	0.9	0.8	1	308.1	36.3	41.3

(c) Profile of example viral items at time t = 360

	dc	**t-dc**	**fo**	**t-fo**	$H(\overrightarrow{\mathbf{D}}^{item}_{pro})$	$H(\overrightarrow{\mathbf{D}}^{item}_{pro,t})$	$H(\overrightarrow{\mathbf{D}}^{item}_{inf})$	$H(\overrightarrow{\mathbf{D}}^{item}_{inf,t})$	**#ni**	**mdu**	**inc**	**t-mdu**
item-x	870	837.7	1.5	1.5	0.9	0.9	1.0	1.0	600	389.2	593.6	435.0
item-y	53,059	447.2	2.1	1.4	0.9	0.9	0.8	0.8	36	20,886.5	208.2	172.5

generates many more propagation instances than user-b and also propagates more diversely (as measured by $H(\overrightarrow{\mathbf{D}}^{pro}_{item,t})$). This suggests that user-a is recently more viral than user-b. The **t-mdu** model therefore reasonably assigns higher virality score to user-a.

Susceptible user example. Similarly, in Figure 4.4 (b), user-c is a news aggregator-like account that actively retweets from Korean music celebrities, and user-d is a sport player account. Historically, user-d has been often infected with items more diversely. Hence, it is reasonable that user-d is assigns a higher susceptibility score by the static models. However, in the current time step (t = 360), user-c is infected with many more items propagated by different users. The **t-mdu** model therefore reasonably assigns higher susceptibility score to user-c.

Viral item example. Lastly, in Figure 4.4 (c), item-x is a hashtag people use to tweet about an entertainment event, and item-y is a hashtag indicating that a Twitter user is willing to follow other users. The figure shows that item-y was highly propagated in the past, while item-x is much more diversely propagated by more users in the recent time step (t = 360). It is therefore reasonable that item-y is assigned much higher virality scores by other models, while item-x is assigned higher virality score by the **t-mdu** model.

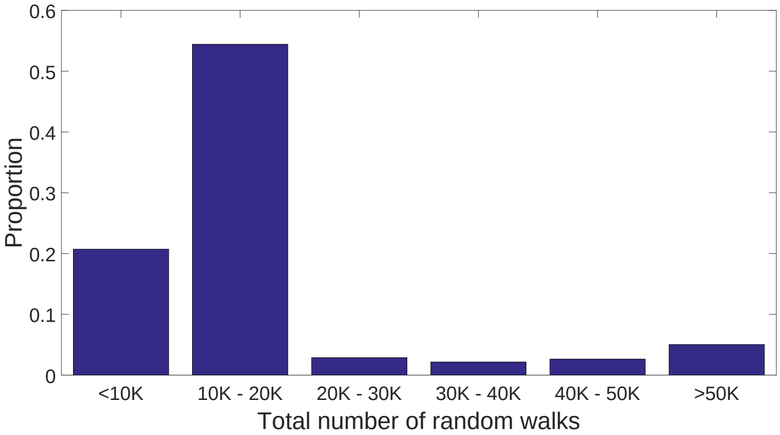


Figure 4.6: Average proportion of random walks need to be re-conducted in updating the propagation rank vector.

4.6.5 Accuracy Evaluation

We now evaluate the accuracy of different models. Since there is no ground truth for viral (or susceptible) users, and viral items, at every time step in the dataset, we use retweets observed in the next time step for accuracy evaluation.

Evaluating Methodology. Prediction of **high likelihood of retweet** and **high retweet count** are two possible tasks to evaluate the models. As the likelihood of retweet requires knowledge about users reading tweets which is usually not available, we employ retweet count prediction in this experiment.

Since we defined propagation based on retweets, we expect that, at each time step, top viral and susceptible users and top viral items are the ones generating most retweets in the next time step. Moreover, since each retweet is jointly contributed by the user and item factors, we evaluate the joint contribution of future retweets by the top viral and susceptible users and top viral items. As baseline models measure only a single user/item factor, we combine them so as to perform the prediction task using three factors together.

At any time step t, for each model or combination of baselines, we count the number of retweets in time step $t+1$ by top K viral users, top K viral items, and/or top K susceptible users ($K \in \{10, 20, \cdots, 100\}$). We then normalize the count by the total number of retweets in the time step $t+1$ to get the proportion of retweets generated by the tops K users/items. The model with higher proportions across the time steps is therefore more accurate in this prediction task.

Results. Figure 4.7 shows the average proportion of retweets in the next time step returned by different models and combinations of baseline models over all time steps, and with different values of K. In the figure, **pc & ic & p** denotes the combination of baseline models where user virality is measured by **pc** model, user susceptibility is measured by **ic** model, and item virality is measured by **p** model. Other combinations of baseline models are named in a similar manner.

The results yield the following consistent observations. Firstly, the combinations of count based baselines (**pc**, **t-pc**, **p** and **t-p**) only outperform the combinations with the proportion based baselines (**fo**, **t-fo**, **vc** and **t-vc**). This suggests that actual numbers of propagation instances are better indicators of user virality and susceptibility and item virality. Secondly, most of temporal models and combinations of temporal baselines outperform the corresponding static models and combinations of static baselines. This shows the effectiveness of the temporal weighting scheme used in the temporal models. Third, our proposed mutual dependency & unbiased models significantly outperform combinations of the baselines in both variants: static and temporal variants. Lastly, as we expected, the incremental model (**inc**) shares similar performance as **t-mdu**, and significantly outperforms all other models and combinations of baselines.

4.6.6 Incremental Model Evaluation

Previous experiments presented above show that the **inc** model is as effective as the **t-mdu** model. We now evaluate the speedup of **inc** compared with the **t-mdu** model, and the amount of work in each incremental update in **inc** model.

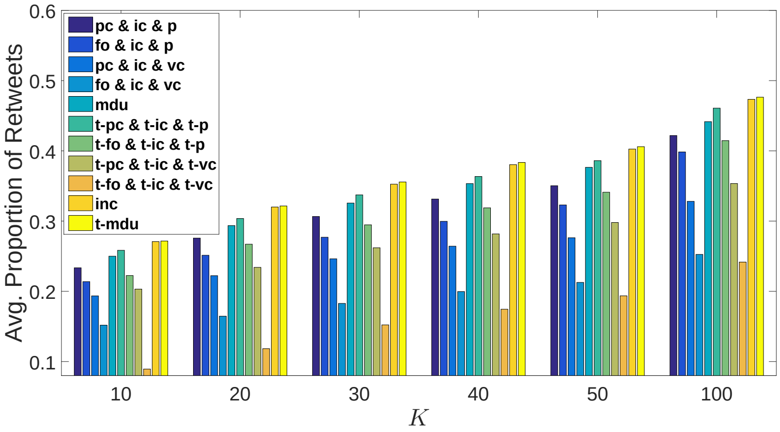


Figure 4.7: Average proportion of retweets in next time step that are generated by either top K viral users, top K susceptible users, or top K viral items returned by different models.

Speedup ratio. We found that on average that the iterative computation method used in **t-mdu** model needs more than 10 iterations to reach convergence. This means that, the computation of the **inc** model is **more than 10 times faster** than that of the **t-mdu** model as **inc** model uses only 1 iteration.

Incremental updating cost. We examine actual computational cost for each time we update the Pagerank approximation. Recall that we do not update the approximation after each single propagation instance comes, but after a short time window of 15 seconds. Figure 4.6 shows the average proportion of the random walks we need to re-conduct in different bins of total number of the walks. The figure clearly shows that, as we expected, the proportion is quite high when the total number of walks is small, and is much lower when the number of walks increases. The fact that the proportion in the first bin (when the total number of walks is less than 10K) is smaller than in the second bin (when the total number of walks is from 10K to 20K) is also reasonable. We found that, in most of the cases in the second bin, the propagation graph is more dense, and hence each user is generally visited by more random walks. Therefore, each new propagate instance requires us to re-conduct more walks. However, in other bins, the propagation graph is larger but less dense, making the proportion of walks we have to re-conduct drops drastically. On average, there is less than 5% of walks need to be re-conducted.

4.7 Chapter Summary

In this chapter, we proposed static, temporal, and incremental user and item factor models for joint modeling of user virality, user susceptibility, and item virality from large propagation data stream. Our models work in more practical settings than many other existing models wherein user-exposure-to-content item is not observed and users may have multiple adoptions and infections with the same item. Our proposed models consider mutual dependencies between factors as well as the unbiasness of the factor scores. We conducted a series of experiments to show that our models are more intuitive and outperforms the baselines. We also showed that the incremental model is much more computationally efficient than the temporal models while still returns similar results.

Chapter 5

Topic-specific Virality and Susceptibility in Content Propagation

In this chapter, we study the problem of modeling user virality, user susceptibility, and content virality specific to topics. We discuss the motivation for this work in Section 5.1. In Section 5.2, we state our research objectives and highlight our contributions. Section 5.3 provides the justifications that the virality and susceptibility factors should be modeled at the topic level. We describe our proposed modeling framework and its associated models in Section 5.4. We present the experiments for evaluating the proposed models on real and synthetic datasets in in Sections 5.5 and 5.6 respectively. Finally, we conclude the chapter in Section 5.7.

5.1 Motivation

Past studies have shown that some topics are viral, e.g., political and entertainment events [202], and disasters [217], etc., while there are also many other non-viral topics [258, 212, 83]. Previous works have also suggested that both user virality and user susceptibility are topic specific: users are

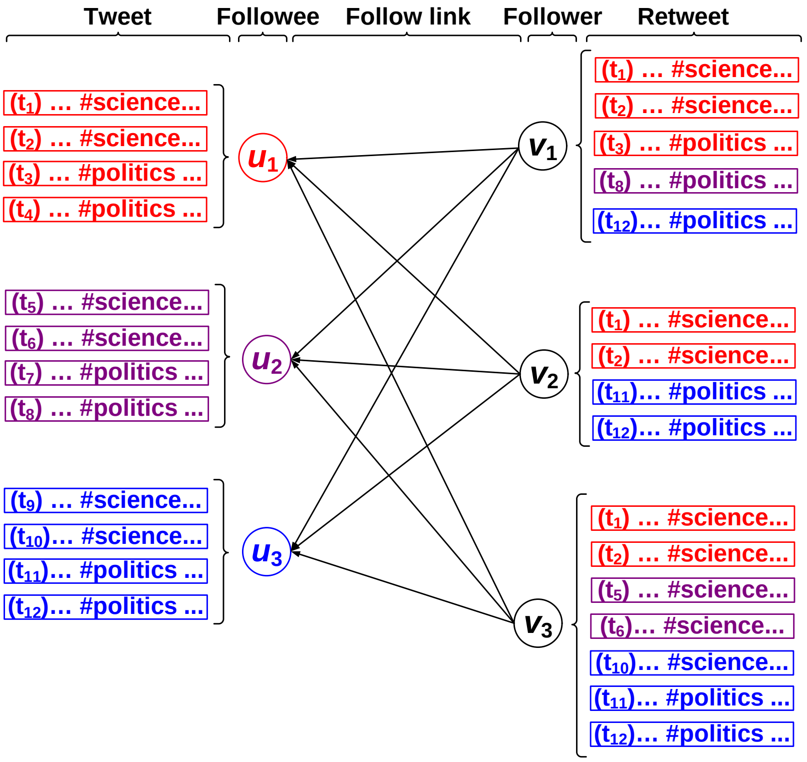


Figure 5.1: Example scenario of topic-specific virality and susceptibility in microblogging

viral/ susceptible on some topics but not viral/ susceptible on other topics [204, 228, 149, 105, 48, 203, 83]. Existing models however ignore content topics (e.g., [74, 33, 206, 187, 48, 87, 22, 84, 1, 11, 226]). Such topic-independent approach may also lead to inaccurate modeling results.

Consider the example scenario of propagation in Twitter shown in Figure 5.1. Here, content are tweets, and they are propagated through retweets. A topic-independent model would conclude that (a) u_1 is more viral than u_3 since the former gets more retweets (i.e., 7) than the latter (i.e., 6), and (b) v_3 is more susceptible than v_1 since the former retweets more than the latter (7 and 5 respectively). However, on *politics* topic, (i) u_3 receives retweets from all the followers, and v_1 retweets all the followees' tweets; while (ii) u_1's tweets are only retweeted by v_1, and v_3 retweets only u_3's tweets. Hence, we may conclude that, for *politics* topic u_3 is more viral than u_1 and v_1 is more susceptible than v_3.

5.2 Research Objective and Contributions

In this work, we aim to jointly model user virality, user susceptibility, and content virality specific to topics. Defined at the topic level, these factors can be used to perform prediction of content propagation more effectively.

To meet the objectives, we have to address a few challenges. Firstly, both content propagation and user-content exposure instances are required for modeling factors specific to topics. However, as mentioned in Chapter 4, we could only observe the adoption and propagation of content by microblogging users, but not their exposure to content. Secondly, microblogging content are known to be very noisy and their topics are not clear. For example, [21, 243] report that as many as 75% of tweets do not carry meaningful topics. Thirdly, the inter-dependencies among the factors remains a challenge since we want to consider the factors at topic level which is not addressed in Chapters 3 and 4. Lastly, the lack of ground-truth information is also still a challenge for the same reason.

We address the first challenge by inferring user-content exposure based on the chronological order in microblogging users' timeline and their following network. To address the second challenge, we devise a multi-steps heuristic method for removing noise and identifying topics of the content, coupling with the state-of-the-art topic model for microblogging content. For the third challenge, we construct a *propagation tensor* representing exposing users - content - exposed users relationship (i.e., who exposed what item(s) to whom), and propose a factorization framework on this tensor to simultaneously derive the three topic-specific behavioral factors. We develop two factorization models base on the framework so as to learn the behavioral factors effectively. Lastly, to evaluate the proposed models, we examine the performance of our models in propagation prediction tasks, comparing them with the state-of-the-art baselines. We also use synthetically generated datasets with known ground-truth to evaluate the models and the learning algorithm.

Our main contributions in this chapter consist of the following.

- We propose a tensor factorization framework, called **V2S** framework, to model an observed content propagation dataset using three behavioral factors, i.e., topic virality, topic-specific user virality, and topic-specific user susceptibility. Within this framework, we develop two factorization methods: *Numerical Factorization Method* and *Probabilistic Factorization Method* to simultaneously measure topics' virality as well as topic-specific users' virality and susceptibility.
- We convert the above constrained factorization problem into a unconstrained optimization which can be solved effectively using gradient descent methods.
- We apply the **V2S** - based factorization models to predict retweets in a large Twitter dataset and show that the models outperform state-of-the-art baseline methods.
- We also conduct extensive experiments on synthetic datasets to verify the effectiveness of our approach in learning the three behavioral factors.

5.3 Empirical Studies

In this section, we conduct an empirical analysis of content propagation on a large dataset collected from Twitter. The methodology used to derive content propagation behavior and topics will be presented. The study will we show that virality and susceptibility should be modeled at topic level.

In microblogging, retweet is the most common form of content propagation. We therefore use retweet to define propagation in the remaining part of this section. That is, *each original tweet m is considered as a content item*, and *we say user v is exposed to m if (a) v follows m's author, and (b) v receives* *and reads* m. Lastly, m *is said to be propagated from its author* u *to* v *if (i)* v *follows* u *and (ii)* v *retweets* m.

5.3.1 Dataset

Our dataset is a large corpus of tweets collected just before the 2012 US presidential election. To construct this corpus, we first manually selected a set of 56 *seed users*. These are highly-followed and politically-oriented Twitter users, including major US politicians, e.g., Barack Obama, Mitt Romney, and Newt Gingrich; well known political bloggers, e.g., America Blog, Red State, and Daily Kos; and political sections of US news media, e.g., CNN Politics, and Huffington Post Politics. The set of users was then expanded by adding all users following at least three seed users so as to get more politics savvy users. Lastly, we crawled the following network among those users and all their tweets posted during the first two weeks of October 2012. This period includes many events related to the 2012 US presidential election, e.g., the national conventions of both democratic and republican parties, and the debates between presidential candidates, etc.. This dataset thus contains both network and content propagation for a large set of Twitter users actively participating US politics during a politically active period. We therefore expect tweets in this dataset to be well read, and highly retweeted.

In Twitter, topics of tweet content change rapidly and so do the user behaviors [123, 132]. We therefore conduct our analysis in a series of sliding time windows derived from the crawled dataset, each within a short duration of time, to examine topics and user behaviors in each window. More precisely, as the crawled dataset spans over 14 days, we divide it into 10 sliding windows: each window spans 5 days, and the sliding step is 1 day. This choice of window size is based on the findings of Yang *et. al.* [236] that most of Twitter content have lifespan of around 5 days. Table 5.1 shows the statistics about the data in each time window. Roughly, in each time window, about 4% of tweets are

Table 5.1: Statistics of the dataset

Time window	#Users	#Tweets	#Retweeted tweets	#Retweets
0	268,676	9,612,207	396,010	1,312,037
1	269,163	9,555,811	391,980	1,309,824
2	268,386	9,362,051	377,298	1,274,902
3	267,898	9,247,465	371,962	1,257,921
4	251,940	7,646,186	284,368	791,901
5	250,559	7,651,155	289,344	802,166
6	252,139	7,941,359	312,342	873,631
7	266,093	9,561,264	414,620	1,419,549
8	265,698	9,363,371	406,117	1,401,437
9	263,262	9,169,674	393,072	1,379,512

retweeted and each of such tweets generates around 3.5 retweets, leading to around 14% of all the tweets are retweets. These numbers are significantly higher than those reported in previous works (e.g., [204, 123]). This confirms that our dataset actually contains tweets that are highly retweeted.

5.3.2 Methodology

Both content propagation and content topics are usually not observable when the microblogging data are crawled. We have therefore devise the methodological steps to infer them as described below.

Determining user-tweet exposure. In Twitter, the latest tweets posted by a user's followees always appear at the top of her timeline. Hence, many tweets may have been missed by the user who does not monitor the timeline closely, and such tweets would never be retweeted. As Twitter API does not reveals the tweets seen by users, we define a time window in which the received tweets will be read. We know that every retweet by a user v comes with a corresponding tweet m that v must have read. We first count the number of other tweets v receives within the duration from the time v receives m to the time v retweets m. Based on this count we estimate N_r the number of tweets a user may read on her timeline whenever she performs a retweet. We found that N_r follows a long tail distribution. For more than 90% of the times,

N_r is not larger than 200. We therefore determine that a user v receives and actually reads through the tweet m, i.e., v is exposed to m, if and only if m is among last 200 tweets posted by v's followees up to the time v makes a retweet. Otherwise, v is considered not exposed to the tweet m.

Topic discovery. We applied TwitterLDA model [258] to automatically identify the topics of every original tweet. This step is conducted for every time window, independently from each others.

We first remove all retweets and non-informative tweets, e.g., tweets generated by third party applications like Foursqure[1] or Instagram[2], etc.. We then remove from remaining tweets all stop words, slang words[3], and non-English phrases. Next, we iteratively filter away words, tweets, and users such that: each word must appear in at least 3 remaining tweets, each tweet contains at least 3 remaining words, and each user has at least 20 remaining tweets. These minimum thresholds are designed to ensure that for each user, tweet, and word, we have enough observations to learn the latent topics accurately.

Figure 5.2 (a) shows the likelihood of the TwitterLDA model in the first time window with respect to the number of topics K varying from 10 to 100. As expected, larger K gives larger likelihood. The quantum of improvement decreases as K increases. Considering both time and space overheads, we set $K = 80$ for the first time window. The number of topics in each of the remaining windows is determined similarly.

Based on the learnt topics and topic distributions of users, we compute the topic distribution of every remaining tweet m with author u as follows.

$$D(m,k) \propto \theta(u,k) \cdot \prod_{w \in m} \phi(k,w) \tag{5.1}$$

where $D(m,k)$ is the probability of topic k in tweet m; $\theta_k(u)$ is the probability of topic k of the author u; and $\phi(k,w)$ is probability of word w given topic k.

[1] https://foursquare.com/

[2] http://instagram.com/

[3] http://en.wikipedia.org/wiki/Slang

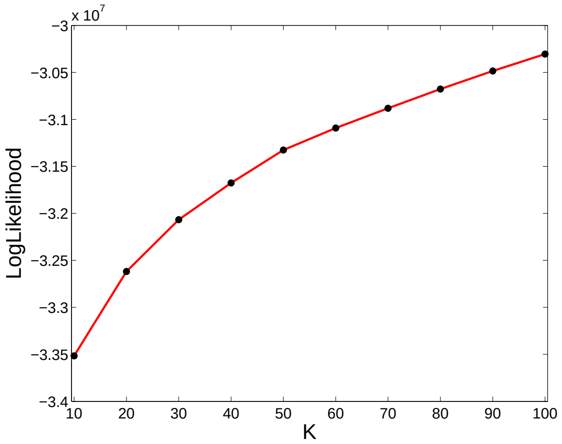


Figure 5.2: Likelihood of the TwitterLDA model in the first time window

Due to the filtering steps above, many tweets are filtered away, and there is only 15% of tweets that are topically modeled by the TwitterLDA model. We therefore expanded the set of modeled tweets as follows. First, we include in the set all the tweets of filtered away users that contain at least 3 remaining words. Then, we compute the topic distribution of each of these tweets using their (remaining) words and the learnt topics, assuming the tweet's author u (who is filtered away) has a uniform distribution over topics (i.e., $\theta_k(u) = 1/K$).

Moreover, as each tweet is a short document, we are not interested in tweets that cover many topics. Instead, we only consider tweets having some dominating topics. To do this, we filter away tweets whose sum of top K_{dom} topic probabilities is less than 0.95. Then, for each of the remaining tweets, we *normalize* topic distribution of the tweet such that sum of K_{dom} highest topic probabilities equals to 1, and all other topics have probability 0. In this study, we set $K_{dom} = 3$. This number is reasonable given that there are some suggestions of assigning only one topic per tweet [258, 243].

Finally, for each time window, we obtained 25% tweets with topic distributions. This is similar to the findings of Balasubramanyan *et. al.* [21] that only about 25% of all tweets are topical tweets.

Table 5.2: Notations used in topic-specific behavioral factors models

$\boldsymbol{\mathcal{M}}$	Set of all content items
$\boldsymbol{\mathcal{M}}_g(u)$	Set of all content items user u generated
$\boldsymbol{\mathcal{M}}_e(v)/\ \boldsymbol{\mathcal{M}}_p(v)$	Set of all content items of user v exposed to/ adopted due to propagation
$p(m)$	Number of time content m is propagated successfully
$D(m,k)$	Probability of topic k in content m's topic distribution
$T(k)/\ T_p(k)$	Global popularity/ propagation popularity of topic k
$T^s(u,k)$	Exposing user-specific popularity of user u for topic k
$T_p^s(u,k)$	Exposing user-specific propagation popularity of user u for topic k
$T^r(v,k)$	Exposed user-specific popularity of user v for topic k
$T_p^r(v,k)$	Exposed user-specific propagation popularity of user v for topic k
(u,v,m)	A propagation observations
δ_{uvm}	Indicator of (u,v,m) observation: $= 1$ if v adopts m, 0 otherwise
$\boldsymbol{\mathcal{O}}$	Set of all propagation observations
$I(k)$	Virality of topic k
I	Topic virality vector
$V(u,k)/\ S(v,k)$	Virality/ susceptibility of user u/ v for topic k
$V(u)/\ S(v)$	Topic-specific virality/ susceptibility vector of user u/ v
$\boldsymbol{\mathcal{V}}_k/\ \boldsymbol{\mathcal{S}}_k$	Set of targeting users for virality/ susceptibility for topic k
$\boldsymbol{\mathcal{V}}/\ \boldsymbol{\mathcal{S}}$	$\bigcup\limits_k \boldsymbol{\mathcal{V}}_k/\ \bigcup\limits_k \boldsymbol{\mathcal{S}}_k$

5.3.3 Empirical Findings

We now present a set of findings about how different topics get propagated (retweeted). In particular, we aim to answer the following questions: (a) Do all topics get equally retweeted? (b) Does a user get relatively same amount of retweets for every topic? and (c) Does a user performs relatively same amount of retweets for every topic?

The main notations used in this chapter are shown in Table 5.2. Like in topic modeling, we conducted the following analysis for every time window independently from the others. Therefore, we exclude the index of time window in the notations for the simplicity in presentation.

5.3.3.1 Topics of tweets and retweets at network level

To compare the likelihood of getting retweeted across topics, in each time window and for each topic k, we derive the relative popularities of topic k among the set of all original tweets and the bag of retweets in the time window. The former is called *global popularity* of the topic k, denoted by $T(k)$, and the later is called *propagation popularity*, denoted by $T_p(k)$. The two popularities

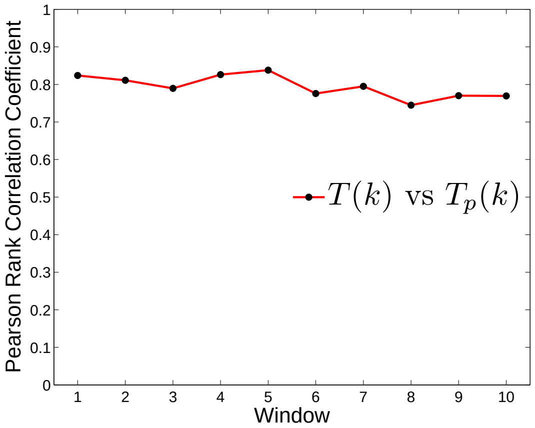


Figure 5.3: Correlation between topics' popularities at network level

are defined based as follows.

$$T(k) = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} D(m,k) \tag{5.2}$$

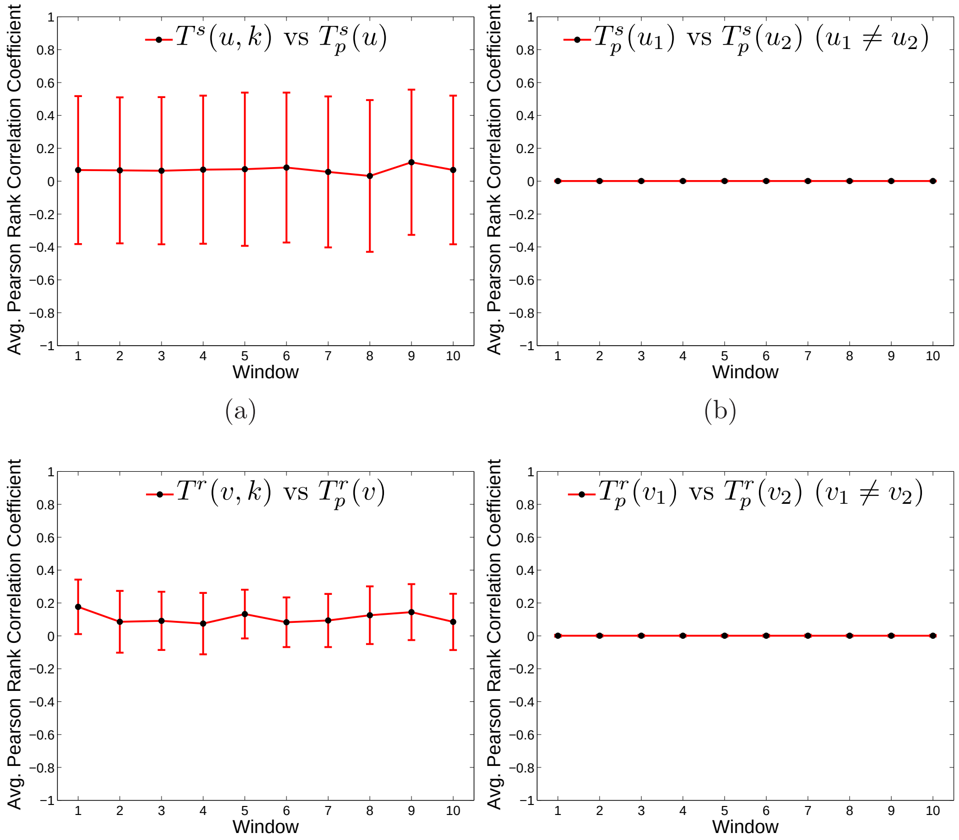
$$T_p(k) = \frac{1}{\sum\limits_{m \in \mathcal{M}} p(m)} \sum_{m \in \mathcal{M}} p(m) \cdot D(m,k) \tag{5.3}$$

where, in each time window, $\mathcal{M}$ is the set of all content items, and $p(m)$ is number of time m is propagated successfully. Since we use tweets and retweets to define content and propagation respectively, $\mathcal{M}$ is the set of original tweets while $p(m)$ is number of m's retweets.

Figure 5.3 shows the Pearson rank correlation coefficient between the two popularities across the time windows. The figure clearly shows that (a) the relative popularity of a topic in the bag of retweets is similar but not the same with the topic's popularity in the set of original tweets; and (b) this observation is consistent across the time windows. This implies that different topics have different likelihood of being retweeted.

5.3.3.2 Topics of tweets and retweets at individual level

On the exposing user side. In each time window, to compare the likelihood of user u getting retweeted for different topics, we compare the relative popularities of each topic k in the set of tweets posted by u, and in the bag-of-



(a)

(b)

(c)

(d)

Figure 5.4: Correlation between topics' popularities at individual level: (a, b) on the author side; and (c, d) on the receiver side

retweets that u got. The former is called *exposing user-specific popularity* of u for topic k, while the latter one is called *exposing user-specific propagation popularity* of u for topic k. The two popularities are denoted by $T^s(u,k)$ and $T_p^s(u,k)$ respectively, and are defined below.

$$T^s(u,k) = \frac{1}{|\boldsymbol{\mathcal{M}}_g(u)|} \sum_{m \in \boldsymbol{\mathcal{M}}_g(u)} D(m,k) \tag{5.4}$$

$$T_p^s(u) = \frac{1}{\sum\limits_{m \in \boldsymbol{\mathcal{M}}_p^{\rightarrow}(u)} p(m)} \sum_{m \in \boldsymbol{\mathcal{M}}_g(u)} [p(m) \cdot D(m,k)] \tag{5.5}$$

where $\boldsymbol{\mathcal{M}}_g(u)$ is the set of content items generated by of u. In this section, $\boldsymbol{\mathcal{M}}_g(u)$ is consist of all u's original tweets.

We compute Pearson rank correlation coefficients between $T^s(u,k)$ and $T_p^s(u,k)$ for each user u, and between $T_p^s(u_1,k)$ and $T_p^s(u_2)$ for each pair of different users u_1 and u_2. Figures 5.4 (a) and (b) show the means and standard deviations of the coefficients across the time windows. The figures clearly show that, for each user, the relative popularities of topics in her bag-of-retweets are different from that popularities in her tweets, and are also different from the popularities in the bag-of-retweets of other users. This implies that (1) the same user has different likelihoods of getting retweeted for different topics, and (2) the same topic has different likelihoods of being retweeted when the topic is mentioned in the tweets generated by different users.

On the exposed user side. Similarly, in each time window, to compare the likelihood of retweeting by user v for different topics, we compute the relative popularities of each topic k in the set of tweets v received and read, and in the set of tweets v retweeted. The former popularity is called *exposed user-specific popularity* of user v for the topic k, and the latter is called *exposed user-specific propagation popularity* of user v for topic k. The two popularities are denoted by $T^r(v,k)$ and $T^r_p(v,k)$ respectively, and are defined below.

$$T^r(v,k) = \frac{1}{|\boldsymbol{\mathcal{M}}_e(v)|} \sum_{m \in \boldsymbol{\mathcal{M}}_e(v)} D(m,k) \tag{5.6}$$

$$T^r_p(v,k) = \frac{1}{|\boldsymbol{\mathcal{M}}_p(v)|} \sum_{m \in \boldsymbol{\mathcal{M}}_p(v)} D(m,k) \tag{5.7}$$

where $\boldsymbol{\mathcal{M}}_e(v)$ and $\boldsymbol{\mathcal{M}}_p(v)$ are the set of content items v has exposed to and the set of all content items v has adopted due to propagation, respectively. In this section, $\boldsymbol{\mathcal{M}}_e(v)$ is consist of original tweets v has received and read, while $\boldsymbol{\mathcal{M}}_p(v)$ is the set of retweets by v.

We compute Pearson rank coefficients between $T^r(v,k)$ and $T^r_p(v,k)$ for each user v, and between $T^r_p(v_1)$ and $T^r_p(v_2)$ for each pair of different users v_1 and v_2. Figures 5.4 (c) and (d) show the means and standard deviations of the coefficients across the time windows. Again, the figure clearly shows that, for each user, the relative popularities of topics in the set of tweets she retweeted are different from that popularities in the set of tweets she received and read, and are also different from the popularities in the set of tweets that other users retweeted. This implies that (1) the same user shows different likelihoods of performing retweet for different topics, and (2) the same topic has different likelihoods of being retweeted when the topic is mentioned in tweets received by different users.

5.4 Content Propagation Modeling Using Topic-specific Behavioral Factors

In this section, we define the topic-specific behavioral factors and present our proposed framework that incorporates all the factors to generate microblogging content propagation data. We also present two models that implement the proposed framework, and describe an algorithm for the models' parameters learning.

5.4.1 Topic-specific Diffusion Behavioral Factors

We now define the following three users' virality/ susceptibility and content's virality specific to topics.

- **Topic virality**: This refers to the ability of a topic to attract propagation. Every topic k is associated to a virality score $I(k) \in [0,1]$ indicating how viral the topic is, i.e. how likely a content about the topic will get propagated.

- **Topic-specific user virality**: This refers to the ability of a user to get her content propagated for a specific topic. We assign to every user u a topic-specific user virality vector $V(u) = \big(V(u,1),\cdots,V(u,K)\big)$ where $V(u,k) \in [0,1]$ for $\forall k = 1,\cdots,K$. For topic k, $V(u,k)$ denotes how viral user u is for the topic, i.e., how likely u gets propagated for her content with topic k.

- **Topic-specific user susceptibility**: This refers to the tendency of a user to adopt content propagated to her for a specific topic. Each user v is associated with a topic-specific user susceptibility vector $S(v) = \big(S(v,1),\cdots,S(v,K)\big)$ where $S(v,k) \in [0,1]$ for $\forall k = 1,\cdots,K$, and $S(v,k)$ indicates how susceptible user v is to topic k, i.e., how likely v adopts a content about the topic k after being exposed to the content.

Note that not all users generate content with a given topic, or have the chances to be exposed to content with the topic from their followees. We therefore may not be able to measure virality and susceptibility for every user-topic pair due to the lack of observation data. Instead, we identify, for each topic k, the subset of users $\boldsymbol{\mathcal{V}}_k$ generating content about the topic, and the subset of users $\boldsymbol{\mathcal{S}}_k$ being exposed to the topic's content. We then measure virality and susceptibility specific to topic k for users in $\boldsymbol{\mathcal{V}}_k$ and in $\boldsymbol{\mathcal{S}}_k$ respectively. We use V to denote the set of all $V(u)$ vectors with $u \in \boldsymbol{\mathcal{V}} = \bigcup_{k=1}^{K} \boldsymbol{\mathcal{V}}_k$, and use S to denote the set of all $S(u)$ vectors with $v \in \boldsymbol{\mathcal{S}} = \cup_{k=1}^{K} \boldsymbol{\mathcal{S}}_k$. Similarly, we use I to denote the vector $\big(I(1),\cdots,I(K)\big)$ of virality scores of all K topics.

5.4.2 The V2S Framework

Our **V2S** framework represents each content propagation observation by a tuple (u,v,m) where m is a content item generated by user u, and exposed to user v. We use a binary variable δ_{uvm} to denote whether v adopts m ($\delta_{uvm} = 1$) or otherwise ($\delta_{uvm} = 0$). We call a propagation observation *positive* or *negative* when $\delta_{uvm} = 1$ and 0 respectively. In **V2S** framework, δ_{uvm} depends on topic-specific virality of u, topic-specific susceptibility of v, and the topics' virality as follows.

Consider a propagation observation (u,v,m), we assume that the likelihood that v adopts m is determined by: **(a)** m's topic distribution $D(m) = \big(D(m,1),\cdots,D(m,K)\big)$; **(b)** u's topic-specific user virality $V(u)$; **(c)** topic virality I; and **(d)** v's topic-specific user susceptibility $S(v)$. Under this assumption, we estimate δ_{uvm} using the dot product of $D(m)$, $V(u)$, I, and $S(v)$. That is,

$$l\big(\delta_{uvm}\big) \propto f\Bigg(\sum_{k=1}^{K}\big[D(m,k)\cdot V(u,k)\cdot I(k)\cdot S(v,k)\big]\Bigg) \tag{5.8}$$

where $f:[0,1]\longrightarrow\mathcal{R}^{+}$ is a non-negative monotonic function; and $l(\delta_{uvm})$ is either (i) an approximation of δ_{uvm}, or (ii) the likelihood of δ_{uvm}, depending on the context. Different forms of the l and f functions give rise to different implementations of the **V2S** framework.

In **V2S** framework, the topics' virality and the users' topic-specific virality and susceptibility can be learnt through solving the following minimization problem.

$$\big(I^{*},V^{*},S^{*}\big) = \underset{I,V,S}{arg.min}\ L\big(I,V,S\big) \tag{5.9}$$

subject to

$$I(k),V(u,k),S(v,k)\in[0,1] \tag{5.10}$$

where and L is the regularized sum-of-loss:

$$L\big(I,V,S\big) = \sum_{(u,v,m)\in\boldsymbol{\mathcal{O}}} loss_{l,f}\big(u,v,m\big) + \alpha\cdot r_1\big(I,V,S\big) + \beta\cdot r_2\big(I,V,S\big) \tag{5.11}$$

where $\boldsymbol{\mathcal{O}}$ is the set of all content propagation observations, and $loss_{l,f}(u,v,m)$ is the loss in estimating δ_{uvm} with respect to the actual form of l and f. The two regularization terms r_1 and r_2 are defined as follows.

$$\begin{aligned} r_1\big(I,V,S\big) = &\sum_{k=1}^{K}\sum_{u\in\boldsymbol{\mathcal{V}}_k}||V(u)-T_p^{s}(u)\cdot\sum_{k=1}^{K}V(u,k)||^2 + \\ &+\sum_{k=1}^{K}\sum_{v\in\boldsymbol{\mathcal{S}}_k}||S(v)-T_p^{r}(v)\cdot\sum_{k=1}^{K}S(v,k)||^2 + ||I-T_p\cdot\sum_{k=1}^{K}I(k)||^2 \end{aligned} \tag{5.12}$$

$$r_2\big(I,V,S\big) = ||I||^2 + \sum_{u\in\boldsymbol{\mathcal{V}}}||V(u)||^2 + \sum_{v\in\boldsymbol{\mathcal{S}}}||S(v)||^2 \tag{5.13}$$

In Equations 5.12 and 5.13, $T_p=\big(T_p(1),\cdots,T_p(K)\big)$ in which $T_p(k)$ is defined in Equation 5.3. $T_p^s(u)$ and $T_p^r(v)$ are similarly formed from $T_p^s(u,k)$s and $T_p^r(v,k)$s which are defined in Equations 5.5 and 5.7 respectively. In Equation 5.12, the term $||V(u) - T_p^s(u) \cdot \sum_{k=1}^{K} V(u,k)||^2$ is the distance between $V(u)$ and $T_p^s(u)$ after weighting the latter by sum of all components of the former. This term ensures that $V(u)$ follows a distribution that is close to $T_p^s(u)$ as we do expect that users should be more viral for topics which they are more likely to get propagated. Similarly, the terms $\sum_{v \in \mathcal{S}} ||S(v) - T_p^r(v) \cdot \sum_{k=1}^{K} S(v,v)||^2$ and $||I - T_p \cdot \sum_{k=1}^{K} I(k)||^2$ ensure that $S(v)$ and I follow distributions that are respectively close to $T_p^r(v)$ and T_p. Lastly, in Equation 5.13, the regularization terms $||I||^2$ and $\sum_{u \in \mathcal{V}} ||V(u)||^2$, and $\sum_{v \in \mathcal{S}} ||S(v)||^2$ are to avoid overfitting.

5.4.3 Factorization Models

We now describe two factorization models built based on the **V2S** framework.

5.4.3.1 Numerical Factorization Model

In this model, we consider $l(\delta_{uvm})$ as an approximation of δ_{uvm}, and f is the identity function. That is,

$$\delta_{uvm} \approx \sum_{k=1}^{K} \left[D(m,k) \cdot V(u,k) \cdot I(k) \cdot S(v,k) \right] \tag{5.14}$$

Given the approximation in Equation 5.14, the loss function $loss_{l,f}(u,v,m)$ is then the squared loss, defined as follows.

$$loss_{l,f}(u,v,m) = \left(\delta_{uvm} - \sum_{k=1}^{K} \left[D(m,k) \cdot V(u,k) \cdot I(k) \cdot S(v,k) \right] \right)^2$$
$$\text{for } \forall (u,v,m) \in \mathcal{O}. \tag{5.15}$$

5.4.3.2 Probabilistic Factorization Model

In this model, we consider $l(\delta_{uvm})$ as the likelihood of δ_{uvm}, and f is a probability distribution. Since $\delta_{uvm} \in \{0,1\}$, we choose f to be the Bernoulli distribution with mean $\mu(u,v,m) = \sum_{k=1}^{K}\big[D(m,k)\cdot V(u,k)\cdot I(k)\cdot S(v,k)\big]$. That is,

$$\begin{aligned}\text{log-likelihood}\big(\delta_{uvm}\big) = \delta_{uvm}\cdot\ln\Bigg(\mu(u,v,m)\Bigg) + \\ + (1-\delta_{uvm})\cdot\ln\Bigg(1-\mu(u,v,m)\Bigg) \quad (5.16)\end{aligned}$$

The loss function $loss_{l,f}(u,v,m)$ is now the negative log-likelihood of δ_{uvm}, defined as follows.

$$\begin{aligned}loss_{l,f}(u,v,m) = -\delta_{uvm}\cdot\ln\Bigg(\mu(u,v,m)\Bigg) - \\ -(1-\delta_{uvm})\cdot\ln\Bigg(1-\mu(u,v,m)\Bigg) \text{ for } \forall(u,v,m)\in\mathcal{O}. \quad (5.17)\end{aligned}$$

5.4.4 Model Learning

Learning algorithm. With respect to the loss defined in Equations 5.15 or 5.17, minimizing $L\big(I,V,S\big)$ as in Equation 5.9 is a constrained alternating convex problem which could only be solved locally, e.g., by gradient based methods. However, due to the conditions in Equation 5.10, we cannot directly apply the gradient descent methods as they are used for unconstrained problems. To deal with the conditions, we employ the following transformation to transform Problem 5.9 into a unconstrained problem.

$$x = h(z) \text{ or } z = h^{-1}(x) \text{ for } \forall x\in[0,1] \quad (5.18)$$

where h is a S-shape continuous monotone map from $\mathcal{R}$ to $[0,1]$, defined as below.

$$h(z) = \frac{1}{2}\times\frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}+\frac{1}{2} \quad (5.19)$$

Now, denote $\mathcal{Z}^{t} = \big(h^{-1}(I(k)),\cdots,h^{-1}(I(K))\big)$, $\mathcal{Z}^{s}(u) = \big(h^{-1}(V(u,1)),\cdots, h^{-1}(V(u,K))\big)$, and $\mathcal{Z}^{r}(v) = \big(h^{-1}(S(v,k)),\cdots,h^{-1}(S(v,K))\big)$, then Problem 5.9

becomes a unconstrained optimization problem with respect to $\mathcal{Z}^t$, $\mathcal{Z}^s(u)$, $\mathcal{Z}^r(v)$ which now can be solved using gradient descent based methods. To do this, we employ the *alternating gradient descent* method. The main idea is to (**1**) perform gradient descent steps by $\mathcal{Z}^s$ directions while keeping $\mathcal{Z}^r$ and $\mathcal{Z}^t$ unchanged, followed by (**2**) performing gradient descent steps by $\mathcal{Z}^r$ directions while keeping $\mathcal{Z}^s$ and $\mathcal{Z}^t$ unchanged, and lastly (**3**) perform gradient descent steps by $\mathcal{Z}^t$ directions while keeping $\mathcal{Z}^s$ and $\mathcal{Z}^r$ unchanged. This process repeats until we reach a predefined maximum number of iterations or when the values converge.

Complexity. The main computational cost in above learning procedure is in evaluation of the regularized sum-of-loss $L\big(I,V,S\big)$. From Equations 5.11, 5.15, and 5.17, we know that the cost includes (1) cost of computing the loss in estimating all propagation observations, and (2) cost of computing the regularization terms. The former is $\mathcal{O}(K_{dom}\cdot|\boldsymbol{\mathcal{O}}|)$ since we normalized topic distribution of tweets so that each tweet has at most K_{dom} topics, and the latter is $\mathcal{O}(K\cdot(2+|\boldsymbol{\mathcal{V}}|+|\boldsymbol{\mathcal{S}}|))$. Hence, the cost of evaluating $L(I,V,S)$ is linear to the number of propagation observations $|\boldsymbol{\mathcal{O}}|$, the number of topics K, and the number of users $|\boldsymbol{\mathcal{V}}|+|\boldsymbol{\mathcal{S}}|$. Our method is therefore scalable to large datasets.

Parallel implementation. We present here an implementation of the above learning algorithm that allows us to quickly evaluate the regularized sum-of-loss $L\big(I,V,S\big)$ and its gradients by parallel computing. We first rewrite the loss function as follows.

$$L\big(I,V,S\big)=\alpha\cdot r_1\big(I,V,S\big)+\beta\cdot r_2\big(I,V,S\big)+\sum_{u\in\mathbf{V}}\left(\sum_{(u,v,m)\in\boldsymbol{\mathcal{O}}_u}loss_{l,f}\big(u,v,m\big)\right)$$

where $\boldsymbol{\mathcal{O}}_u$ is the set of all propagation observations wherein u is the sender, i.e., $\boldsymbol{\mathcal{O}}_u=\{(u,v,m):(u,v,m)\in\boldsymbol{\mathcal{O}}\}$. As suggested by the equation above, to evaluate $L\big(I,V,S\big)$, we can use multiple child processes, each corresponding to a sender u, to compute $\sum\limits_{(u,v,m)\in\boldsymbol{\mathcal{O}}_u}loss_{l,f}\big(u,v,m\big)$ simultaneously. We then use a master process to compute $\alpha \cdot r_1\big(I,V,S\big) + \beta \cdot r_2\big(I,V,S\big)$ and aggregate results returned by the child processes.

Similarly, the computation of gradient of $L\big(I,V,S\big)$ by a direction is independent from those of all other directions (regardless of the variable transformation as in Equation 5.19). Hence, the gradient of $L\big(I,V,S\big)$ by $\mathcal{Z}^t$, $\mathcal{Z}^s(u)$, and $\mathcal{Z}^r(v)$ directions can also be computed simultaneously using multiple child processes, each corresponding to a direction $h^{-1}\big(I(k)\big)$, $h^{-1}\big(V(u,k)\big)$, or $h^{-1}\big(S(v,k)\big)$.

In our implementation, in evaluating $L\big(I,V,S\big)$, we build a process pool, and submit a process for computing $\sum_{(u,v,m)\in\mathcal{O}_u} loss_{l,f}(u,v,m)$ to the pool for each sender u. At any time, a fixed number $\mathbf{P}$ of the pool's processes are running. In the ideal case, we can reduce the running time of $L\big(I,V,S\big)$ to $\mathbf{P}$ times. Similarly, we use process pool to reduce the running time in computing the gradients and updating the variables.

5.5 Experiments on a Real Dataset

In this section, we evaluate and compare our proposed methods with some baseline methods in future propagation prediction task. Again, we use the Twitter dataset described in Section 5.3.1.

To deal with the dynamic of topics and the propagation factors, our dataset is divided into 10 consecutive sliding time windows, each spans 5 days. Since we want to examine different models in predicting propagation for the future content, we conduct the same experiments for the time windows independently. This also allows us to examine the consistency of the predictive power of models across time. Like in Section 5.3, we use original tweets as content, and retweets as content propagation. That is, for each time window, we *train the models using data from first 4 days of the window*, and use the models to *predict retweets for tweets posted in the last day of the window*.

5.5.1 Data Preprocessing

Topic discovery. For each time window, we first apply TwitterLDA model on the set of all tweets posted in the first 4 days of the window. We use the same pre- and post-processing steps as in Section 5.3.2 for learning topics of the tweets. We then use the learnt topic model to infer topics of the tweets posted in the last day of the window.

For clarity, for each time window, we call the tweet m a *training tweet* if (i) m is posted in the first 4 days of the window, and (ii) m is topically modeled. Similarly, we call the tweet m' a *test tweet* if (i) m' is posted in the last day of the window, and (ii) m' is topically modeled.

Training and test sets. We first apply the same steps presented in Section 5.3.2 to determine user-tweet exposure and identify all propagation observations. We then construct the training and test sets of every time window as follows.

As mentioned in Section 5.4.1, for each time window and each topic k, we only can measure user virality specific to topic k for a subset of users $\boldsymbol{\mathcal{V}}_k$ tweeting about the topic, and measure user susceptibility specific to topic k for a subset of users $\boldsymbol{\mathcal{S}}_k$ who are exposed to tweets about the topic. We therefore have to determine $\boldsymbol{\mathcal{V}}_k$ and $\boldsymbol{\mathcal{S}}_k$ for every topic k. To do this, we first set $\boldsymbol{\mathcal{V}}_k$ and $\boldsymbol{\mathcal{S}}_k$ to be the set of all users in our dataset. Then, to ensure that we have sufficient observations for each user and each topic, we iteratively: (**a**) remove from $\boldsymbol{\mathcal{V}}_k$ users who have less than 5 training tweets about the topic k that are read by users in $\boldsymbol{\mathcal{S}}_k$; and (**b**), remove from $\boldsymbol{\mathcal{S}}_k$ users who either have no retweet on the training tweets posted by users in $\boldsymbol{\mathcal{V}}_k$, or read less than 5 training tweets about the topic k that are posted by users in $\boldsymbol{\mathcal{V}}_k$. The training set of the time window then includes all retweet observations (u, v, m) wherein $u \in \boldsymbol{\mathcal{V}}$, $v \in \boldsymbol{\mathcal{S}}$, and m is a training tweet posted by u. Lastly, the test set of the time window includes all retweet observations (u, v, m') wherein $u \in \boldsymbol{\mathcal{V}}$, $v \in \boldsymbol{\mathcal{S}}$, and m' is a test tweet posted by u.

Table 5.3: Statistics of the experimental dataset (**ExpDB**).

Time window	$\lvert\boldsymbol{\mathcal{V}}\rvert$	Avg. $\lvert\boldsymbol{\mathcal{V}}_k\rvert$	$\lvert\boldsymbol{\mathcal{S}}\rvert$	Avg. $\lvert\boldsymbol{\mathcal{S}}_k\rvert$	#observation in training set		#observation in test set	
					all	positive	all	positive
1	6,795	664.95	26,295	8,475.65	8,647,038	75,161	1,643,727	11,382
2	6,786	677.85	26,280	9,188.06	8,985,206	76,127	1,044,329	7,050
3	6,063	607.79	24,391	8,001.73	7,717,675	67,261	921,216	6,525
4	5,823	557.54	23,072	7,010.48	7,022,667	62,576	1,215,506	8,617
5	4,107	397.25	10,701	3,624.50	3,300,547	25,143	1,022,287	6,961
6	3,596	361.89	8,990	3,361.96	2,687,635	20,722	880,724	6,004
7	4,372	444.04	11,396	4,342.80	3,719,318	28,099	1,152,191	8,129
8	4,579	487.23	12,763	5,357.58	4,631,836	33,262	2,406,220	17,618
9	6,752	703.26	28,625	9,522.31	10,208,491	90,075	1,086,309	7,806
10	6,540	648.53	27,029	8,786.13	8,980,865	80,957	1,130,862	8,751

Table 5.3 shows the statistics of the final dataset, called **ExpDB** dataset, which has much fewer users than the original dataset due to the different filtering criteria. Nevertheless we still have a large number of retweet observations. The table also shows that (i) the training and test sets have similar positive observation rates across the time windows, and (ii) in all the time windows, **ExpDB** is highly imbalanced with less than 1% positive observations. This makes the prediction task much more difficult.

5.5.2 V2S-based Models & Parameter Settings

We evaluate both two models presented in Section 5.4.3, i.e., **V2S**-based numerical factorization model and **V2S**-based probabilistic factorization model. We denote the former by $\mathbf{V2S}_F$, and the latter by $\mathbf{V2S}_B$.

In learning the models by alternating gradient descent, we found that the converged measure values could be obtained within 50 alternating iterations, each iteration includes 20 gradient descent steps. The control parameters α, and β are also set through empirical evaluation on a large set of tuples of values. We found that parameter set $\alpha = 10^{-4}$ and $\beta = 1$ gives the best performance. This parameter setting is reasonable as $V(u,k)$ and $S(v,k)$ affect only a subset of retweet observations where u and v are involved respectively; but in contrast, we have much fewer variables $I(k)$ that affect a much larger set of retweet observations (where the tweets are about topic k). Hence, I should be regularized with larger a weight than that of V and S.

5.5.3 Prediction Tasks & Evaluation Metrics

We examine the performance of different methods in the following retweet prediction tasks.

Global retweet prediction. In this task, we aim to predict positive retweet observations among all the observations in the test set, regardless of the users in the observations.

For this task, for each retweet prediction method, we generate a ranking of observations in the test set based on the likelihood of retweet returned by the method. We then construct a Precision-Recall (PR) curve from the test set and the ranking, and measure the area under the PR curve (**AUPRC**). Methods with the higher **AUPRC** are the better.

Personalized retweet prediction. In this task, given a receiver v, we aim to predict tweets that v retweets among all the tweets in the test set that v receives.

In this task, for each retweet prediction method and for each receiver v, we generate a ranking of v's observations in the test set based on the likelihood of retweet returned by the method. We then construct a PR curve from the v's test observations and the ranking. Lastly, we compute the average area under all the receivers' PR curves (**Avg. AUCPR**). Methods with the higher **Avg. AUPRC** are the better.

5.5.4 Comparison with Baselines

We first compare our proposed **V2S**-based methods with the following baselines for diffusion behavioral factors.

5.5.4.1 Baselines

We choose **FanOut** and **FanIn** as baseline for user virality and susceptibility respectively. In our context, the topic-specific FanOut $f_o(u,k)$ of sender u for topic k is defined as the ratio between propagation popularity $T_p^s(u,k)$ of u for topic k (defined in Equation 5.5), and tweet popularity $T^s(u,k)$ of u for topic k (defined in Equation 5.4)

$$f_o(u,k) = \begin{cases} \dfrac{T_p^s(u,k)}{T^s(u,k)} & \text{if } T^s(u) > 0 \\ 0 & \text{otherwise} \end{cases}$$

Similarly, the topic-specific FanIn $f_i(v,k)$ of receiver v for topic k is defined as the ratio between propagation popularity $T_p^r(v,k)$ of v for topic k (defined in Equation 5.7), and tweet popularity $T^r(v,k)$ of v for topic k (defined in Equation 5.6)

$$f_i(v,k) = \begin{cases} \dfrac{T_p^r(v,k)}{T^r(v,k)} & \text{if } T^r(v,k) > 0 \\ 0 & \text{otherwise} \end{cases}$$

Lastly, we use the following baselines for virality of topic k.

- **Global popularity** $T(k)$ as defined in Equation 5.2
- **Propagation popularity** $T_p(k)$ as defined in Equation 5.3
- **Viral coefficient** $vc(k)$ defined as the average number of times an original tweet about topic k is propagated (retweeted). That is,

$$vc(k) = \frac{1}{|\{m \in \boldsymbol{\mathcal{M}} : D(m,k) > 0\}|} \sum_{m \in \boldsymbol{\mathcal{M}}} D(m,k) \cdot p(m)$$

As above baselines measure only a single user/topic factor, we combine them to the following retweet prediction methods using three factors together.

- **FanOut & Global popularity & FanIn**: The likelihood $l_{gp}(u,v,m)$ that $\delta_{uvm} = 1$ is defined as follows.

$$l_{gp}(u,v,m) = \sum_{k=1}^{K} \left[D(m,k) \cdot f_o(u,k) \cdot T(k) \cdot f_i(v,k) \right]$$

- **FanOut & Propagation popularity & FanIn**: The likelihood $l_{pp}(u,v,m)$ that $\delta_{uvm} = 1$ is defined as follows.

$$l_{pp}(u,v,m) = \sum_{k=1}^{K} \left[D(m,k) \cdot f_o(u,k) \cdot T_p(k) \cdot f_i(v,k) \right]$$

- **FanOut & Viral coefficient & FanIn**: The likelihood $l_{vc}(u,v,m)$ that $\delta_{uvm} = 1$ is defined as follows.

$$l_{vc}(u,v,m) = \sum_{k=1}^{K} \left[D(m,k) \cdot f_o(u,k) \cdot vc(k) \cdot f_i(v,k) \right]$$

5.5.4.2 Performance Comparison

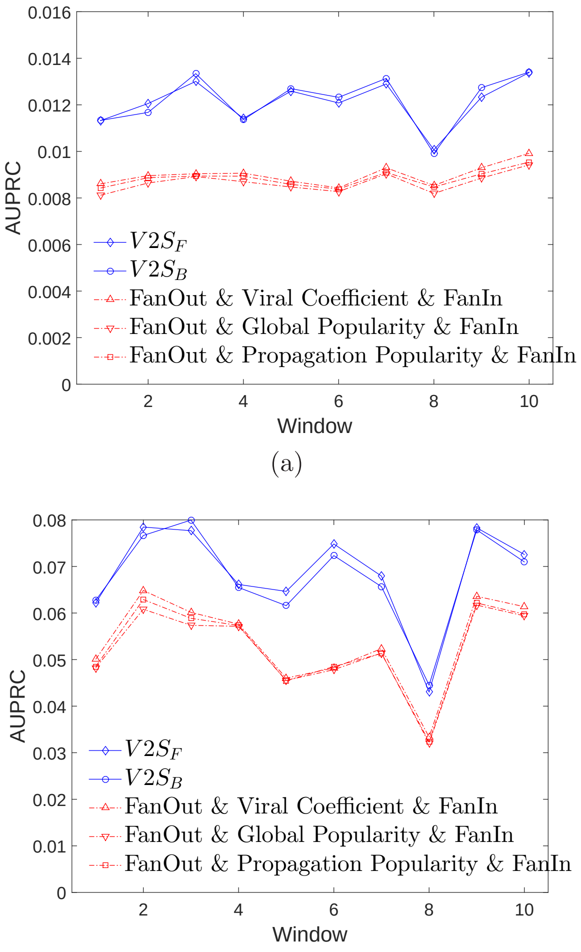
Figure 5.5 (a) shows the performance of **V2S**-based models and other baseline models in global retweet prediction task, while Figure 5.5 (b) shows the models' performance in personalized retweet prediction task. The figures clearly show that (i) the two **V2S**-based models have similar results while the three baselines models have similar results, and (b), across time windows, the **V2S**-based models consistently outperform the baseline models significantly.

5.5.5 Comparison with Content-based Baselines for Retweet Prediction

5.5.5.1 Baseline Models

In this section, we compare **V2S**-based methods with methods specially designed for retweet prediction which can be viewed as a kind of recommendation task.

Since we want to predict retweets on new tweets, which are not used in training the models, the prediction tasks are out-matrix recommendation. However, existing retweet prediction methods (e.g.,[245, 36, 233, 61]) and other simple item- and user-based methods (e.g., matrix factorization, or top similar



(a)

(b)

Figure 5.5: Performance of different models for diffusion behavioral factors in (a) global retweet prediction task, and (b) personalized retweet prediction task.

users and items, etc.) are only for in-matrix recommendation, and hence are not applicable. We therefore compare our proposed **V2S**-based methods with the following content-based baseline models for the retweet prediction tasks.

$\mathbf{TB_r}$ model: The likelihood that $\delta_{uvm} = 1$ depends on topic of m, and topics where v is more likely to adopt due to propagation (retweet).

$$TB_r\big(u, v, m\big) = \sum_{k=1}^{K} \big[D(m,k) \cdot T_p^r(v,k)\big]$$

$\mathbf{TB_{sr}}$ model: The likelihood that $\delta_{uvm} = 1$ depends on topics of m, topics where u is more likely to get propagated (retweeted), and topics where v is more likely to retweet.

$$TB_{sr}\big(u,v,m\big) = \sum_{k=1}^{K}\big[D(m,k)\cdot T_p^s(u,k)\cdot T_p^r(v,k)\big]$$

TB$_{\mathbf{tr}}$ model: The likelihood of $\delta_{uvm} = 1$ depends topics of m, topics that are more likely to be retweeted by all users, and topics where v is more likely to retweet.

$$TB_{tr}\big(u,v,m\big) = \sum_{k=1}^{K}\big[D(m,k)\cdot T_p(k)\cdot T_p^r(v,k)\big]$$

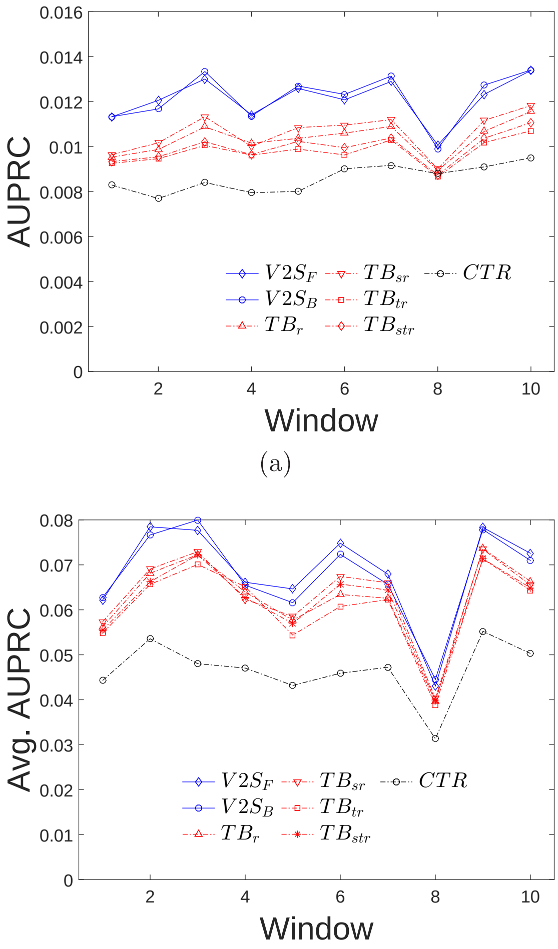
TB$_{\mathbf{str}}$ model: The likelihood that $\delta_{uvm} = 1$ depends topics of m, topics where u is more likely to get retweeted, topics that are more likely to be retweeted by all users, and topics where v is more likely to retweet.

$$TB_{str}\big(u,v,m\big) = \sum_{k=1}^{K}\big[D(m,k)\cdot T_p^s(u,k)\cdot T_p(k)\cdot T_p^r(v,k)\big]$$

Collaborative Topic Regression (CTR) model [222]: This model combines collaborative filtering data with content-based features to perform recommendation tasks. Similar to our proposed methods, **CTR** is solely based on hidden user and content characteristics, and therefore is a suitable baseline. In applying **CTR**, we set the number of topics to the same with that of TwitterLDA model (see Section 5.3.2).

5.5.5.2 Performance Comparison

Figure 5.6 (a) shows the performance of **V2S**-based methods and other content-based baseline methods in global retweet prediction task, while Figure 5.5 (b) shows the models' performance in personalized retweet prediction task. Among the baseline methods, **TB$_{\mathbf{r}}$** and **TB$_{\mathbf{sr}}$** outperform the others in both tasks. This suggests that user specific retweetable topics give a stronger retweet prediction than globally retweetable topics. The fact **CTR** performs worse can be explained by **CTR** suffering from noise as the model infers tweet



(a)

(b)

Figure 5.6: Performance of different retweet recommendation models in (a) global retweet prediction task, and (b) personalized retweet prediction task.

topics and user preference simultaneously, while other methods does not since we employ the topic normalization step (see Section 5.3.2). Again, the figures clearly show that, across time windows, the **V2S**-based methods consistently outperform the content-based baseline models significantly.

5.5.6 Case Studies

We present here case studies to illustrate how the **V2S**-based methods work differently than the baselines.

Viral topic example. Table 5.4 (a) shows the profiles of topics having significantly different scores by different topic virality models. For each topic,

[4] https://en.wikipedia.org/wiki/United_States_presidential_election_debates,_2012

Table 5.4: Case examples of(a) viral topics , (b) viral users, and (c) susceptible users.

(a) Profile of example viral topics at time window 4

Topic Id	Topic Label	#On-topic tweets	#On-topic retweet observations	#On-topic positive observations (rate)	Proportion of retweets		Global popularity	Propagation popularity	Viral coefficient	Virality	
					of top 1% retweeted senders	by top 1% retweeting receivers				by $\mathbf{V2S}_F$	by $\mathbf{V2S}_B$
2	Romney at the 1st presidential debate[4]	15,769	244,604	2,144 (0.9%)	19.4%	5.2%	**0.08**	**0.10**	0.05	0.18	0.10
12	Obama at the 1st presidential debate[4]	14,061	223,985	2,086 (0.9%)	17.4%	5.0%	0.07	**0.10**	0.04	0.86	0.27
71	Unemploy- -ment rate	6,427	183,004	2,211 (1.2%)	23.4%	4.9%	0.03	0.09	**0.14**	0.79	0.26
41	"Big bird" icon in 2012 presidential election campaigns	4,428	55,725	573 (1.0%)	14.3%	2.6%	0.01	0.03	0.07	**0.99**	**0.87**

(b) Profile of example viral users at topic 2 (*Romney at the 1st presidential debate*[4])

user	#On-topic tweets	#On-topic retweet observations	#On-topic positive observations (rate)	Proportion of retweets by top 10% retweeting receivers	FanOut	Virality	
						by $\mathbf{V2S}_F$	by $\mathbf{V2S}_B$
rolandsmartin	50	5,618	57 (1.0%)	26.3%	**1.3**	0.24	0.34
mmfa	9	2,198	47 (2.1%)	14.9%	1.1	**0.65**	**0.96**

(c) Profile of example susceptible users at topic 2 (*Romney at the 1st presidential debate*[4])

user	#On-topic received tweets	#On-topic retweet observations	#On-topic positive retweet observations	Proportion of retweets of the top retweeted sender	FanIn	Susceptibility	
						by $\mathbf{V2S}_F$	by $\mathbf{V2S}_B$
susieq68old	22	179	10 (3.57%)	50.0%	2.13	0.54	0.76
treecia73	16	104	8 (7.69%)	25.0%	1.23	**0.68**	**0.99**

the topic's label is manually assigned based on its representative words, and further insights from its top tweets. A topic's top words are the words having the highest probabilities given the topic, and the topic's top tweets are the tweets having the lowest perplexities given the topic. Also, for each topic k, we select a set of tweets with the normalized probability of topic k (see Section 5.3.2) is at least $\theta = 0.5$, and call them the *on-topic tweets* of topic k. The table shows that topic 2 (*Romney at the 1st presidential debate*[4]), topic 12 (*Obama at the 1st presidential debate*[4]), and topic 71 (*Unemployment rate*) are more popular and have more retweets than topic 41 (*"Big bird" icon in 2012 election campaigns*). However, the three formers have significantly higher proportions of retweets by their top 1% retweeted senders/retweeting receivers than those of the latter. This suggests that topics 2, 12, and 71's retweets are mostly due to their top viral senders and/or top susceptible receivers. Hence, it is reasonable that topic 41 is assigned much higher virality scores by $\mathbf{V2S}_F$ and $\mathbf{V2S}_B$ models.

Viral user example. Similarly, Table 5.4 (b) shows the profiles of two users having most number of retweets for topic 2 (*Romney at the 1st presidential debate*[4]). The user *rolandsmartin* has more retweets for topic 2 than the user *mmfa*. However, on the topic, *rolandsmartin* has lower retweeting rate. Also, the table shows that *rolandsmartin*'s proportion of retweets by top 10% of her retweeting receivers is significantly higher than that of *mmfa*. This suggests that *rolandsmartin*'s retweeting users are more susceptible at topic 2 than those of *mmfa*. It is therefore reasonable that *mmfa* is assigned much higher virality scores by $\mathbf{V2S}_F$ and $\mathbf{V2S}_B$ models.

Susceptible user example. Lastly, Table 5.4 (c) shows the profiles of two users retweet the most for topic 2 (*Romney at the 1st presidential debate*[4]). The user *susie68old* retweets more for the topic than the user *treecia73*. However, *susie68old* has lower retweeting rate for the topic. Also, on topic 2, the table shows that *susie68old*'s proportion of retweets by her top retweeted senders is significantly higher than that of *treecia73*. This suggests that *susie68old*'s retweets are mostly due to a viral sender. $\mathbf{V2S}_F$ and $\mathbf{V2S}_B$ models therefore reasonably assign higher susceptibility scores to *treecia73*.

5.6 Experiments on Synthetic Datasets

Since real datasets do not have ground-truth information on the virality and susceptibility factors, it is impossible to evaluate the accuracy and effectiveness of the models in recovering the factors using the datasets. We address this by conducting experiments on synthetically generated datasets.

5.6.1 Synthetic Data Generation

Generating the user network. We generate a follow network of N users whose in- and out-degrees are at least d_min and have power law distributions with exponent α as follows. We first sample a degree sequence of N nodes from the power law distribution. We then sample links for the nodes using the *expected degree model* [42] with the generated degree sequence. Lastly, for each node having less than d_{min} incoming links, we sample more incoming links for the node using the same probabilities as in the previous step until it gets d_{min} incoming links. Similarly, we sample more outgoing links for the nodes until each has at least d_{min} outgoing links.

Generating the tweets. Given the number of topics K and the number of topics dominating each tweet $K_{dom} < K$, we generate the set of tweets for each user as follows. First, we sample a topic distribution for each user so that the distribution is totally skewed to 10% of the K topics. This skewness is to make each user's tweets focus on only some topics and hence, for each topic the user tweets about, we have enough number of retweet observations to learn her virality for the topic. Then, the number of tweets of each user is uniformly drawn from the range $[n_{min}^{tweet}, n_{max}^{tweet}]$. To generate topic distribution for a tweet of user u, we sample the tweet's main topic from u' topic distribution. We then assign a probability of 0.9 for this main topic. Lastly, we also randomly choose other $K_{dom} - 1$ other (dominating) topics of the tweet, and randomly assign probabilities for these chosen topics so that the probabilities sum up to 0.1.

Generating the ground-truth scores. We randomly choose a small number of topics, let say K_{viral} = 10% of K, to be viral topics. These topics have virality scores randomly uniformly drawn from $[1 - \epsilon, 1)$ for a small value of the so called *score width* ϵ, while the remaining topics have scores uniformly drawn from $[0, \epsilon)$. For each topic, we randomly choose a small number of users having at least one tweet about the topic, let say N_{viral} = 2% of N, to be viral users at the topic. For each user u and each topic k, if u is viral at k, the virality score of u at k is uniformly drawn from $[1 - \epsilon, 1)$. Otherwise, the score is uniformly drawn from $[0, \epsilon)$. Similarly, for each topic, we also choose a small number of users receiving at least one tweet about the topic, let say $N_{susceptible}$ = 10% of N, to be susceptible users at the topic. For each user v

and each topic k, if v is susceptible at k, the susceptibility score of v at k is uniformly drawn from $[1-\epsilon, 1)$. Otherwise, the score is uniformly drawn from $[0, \epsilon)$.

Generating the retweet observations. Now that we have generated the following network, the set of tweets by each users, it is straight forward to determine which users receive which tweets of other users: v receives tweets from u if v follows u. For simplicity, we assume that v reads all the tweets she receives. Hence, we define as retweet observations all the tuples of (u, m, v) where: (i) user v follows user u, and (ii) m is a tweet of user u. A retweet observation (u, m, v), is assigned to be a positive observation (i.e., v retweets m) with the probability $prob(u, m, v)$ computed as follows.

$$prob(u, m, v) = \sum_{k=1}^{K} g_D(m, k) \frac{g_V(u, k) + g_I(k) + g_S(v, k)}{3}$$

wherein, $g_D(m, k)$ is the probability of topic k in tweet m that is generated in the previous step. Similarly, $g_V(u, k)$, $g_I(k)$, and $g_S(v, k)$ are ground-truth virality of user u for topic k, virality of topic k, and susceptibility of user v for topic k as generated previously.

5.6.2 Performance Comparison

We now evaluate our proposed **V2S**-based methods and other baselines in recovering ground-truth topic-specific virality and susceptibility using the synthetic datasets. Similar to experiments in Section 5.5.4, we use **FanOut** and **FanIn** as baselines for user virality and susceptibility respectively, and use **Tweet popularity**, **Retweet popularity**, and **Viral coefficient** as baselines for topic virality.

We generated synthetic datasets with different number of users N, number of topics K, and score width ϵ parameter settings, while fixing $\alpha = 2.5$, $d^i_{min} = d^o_{min} = 3$, $n^{tweet}_{min} = 10$, $n^{tweet}_{min} = 100$, $K_{dom} = 3$, $K_{viral} = 10\%$ of K, $N_{viral} = 2\%$

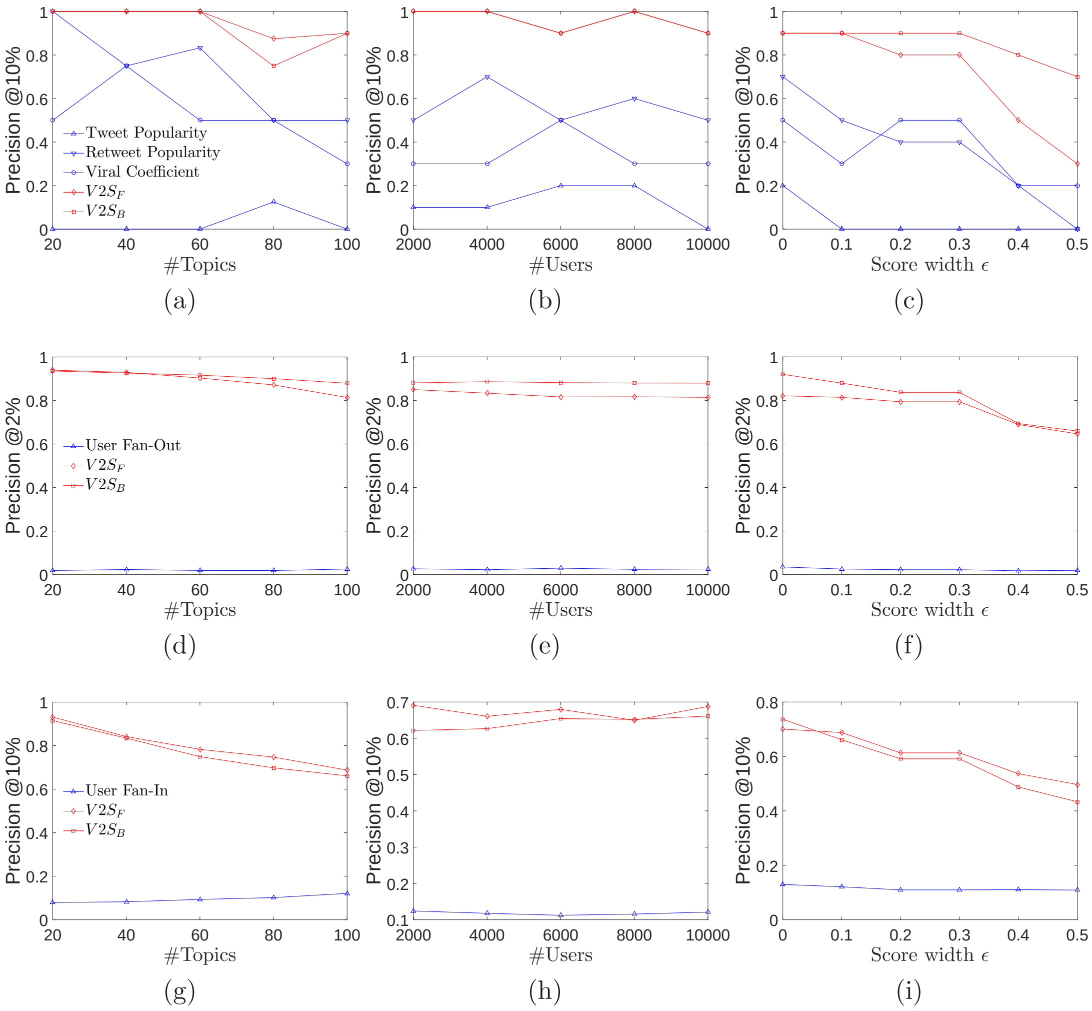


Figure 5.7: Performance of different models in experiments with synthetic datasets

of N, and $N_{susceptible}$ = 10% of N. For each dataset instance and each model, we rank topics by their virality scores produced by the model and select the top scored 10% topics as the predicted viral topics and denote the set by $\mathcal{T}_p$. The precision@10% of the model for topic virality is then defined by $\frac{|\mathcal{T}_p \cap \mathcal{T}_g|}{|\mathcal{T}_g|}$ where $\mathcal{T}_g$ is the set of viral topics in the ground truth. For each topic k, and for each user virality model, the model's precision@2% of topic-specific user virality for topic k is similarly defined, and its precision@2% across topics is computed by averaging the precision from all topics. Lastly, for each user susceptibility model, we compute the model's precision@10% across topics in the similar way.

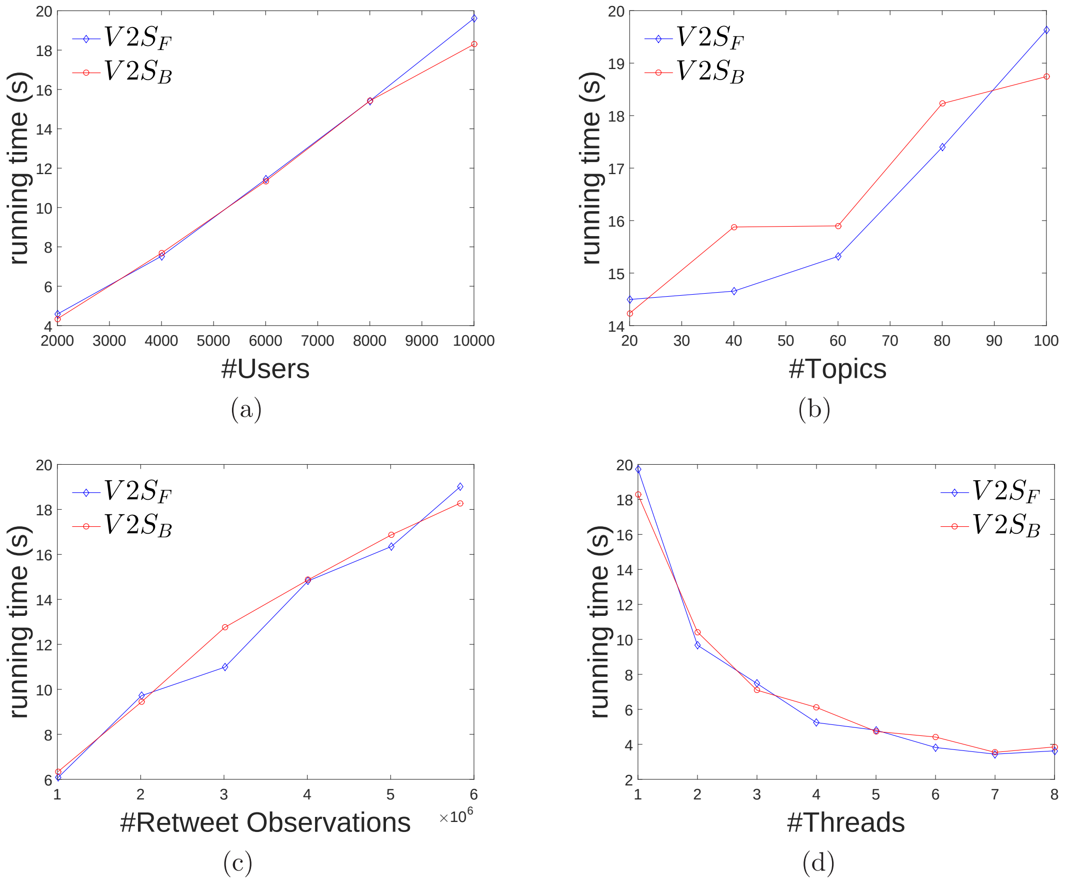
Figures 5.7 (a), (d) and (g) show the precision@10% of topic virality models, precision@2% of user virality models, and precision@10% of user susceptibility models as we varies K from 10 to 100, keeping $N = 10,000$ and $\epsilon = 0.1$. The figures show that the **V2S**-based models significantly outperform other models. All models demonstrate decreasing precision as K increases. They however still outperform the random selection significantly.

Similarly, Figures 5.7 (b), (e) and (h) show the precision@10% of topic virality models, precision@2% of user virality models, and precision@10% of user susceptibility models as we varies N from 1000 to 10,000, keeping $K = 100$ and $\epsilon = 0.1$. Figures 5.7 (c), (f) and (i) show the precisions as we varies ϵ from 0.1 to 0.5, keeping $K = 100$ and $N = 10,000$. Again, all the models demonstrate decreasing precision as N and ϵ increases though still outperform the random selection significantly; and the **V2S**-based models significantly outperform other models.

5.6.3 Scalability

We theoretically analyse the complexity of our learning algorithm for **V2S**-based models and describe a parallel implementation in Sections 5.4.4. We now empirically examine the running time of the algorithm and the efficacy of the implementation.

Running time. Figure 5.8 (a) shows the running time of **V2S**-based models in one alternating iteration as we varies N from 1000 to 10,000, keeping $K = 100$. Similarly, Figure 5.8 (b) shows the running time as we varies K from 20 to 100, keeping $N = 10,000$, and Figure 5.8 (c) shows the running time as we varies number of retweet observations $|\mathbf{O}|$ from 1 million to 10 millions, keeping $K = 100$ and $N = 10,000$. In all these three cases, we keep $\epsilon = 0.1$. The figures clearly show that the running time of **V2S**-based models are linear to the number of users, the number of topics, and the number of retweet observations. This verifies the learning algorithm's theoretical complexity, and shows its scalability.



(a)

(b)

(c)

(d)

Figure 5.8: Running time of the **V2S**-based models in different settings of the number of (a) users, (b) topics, (c)retweet observation, and (d) parallel threads.

Efficacy of the parallel implementation. Figure 5.8 (d) shows the running time of **V2S**-based models in one alternating iteration as we varies the number of parallel processes from 1 to 8, keeping number of retweet observations $|\mathcal{O}| = 10$ millions, $K = 100$, and $N = 10,000$. The figure shows that the larger the number of parallel processes used **P** results in less running time, and the amount of improvement decreases as **P** increases. This shows the efficacy of our parallel implementation. The fact that the running time even increases slightly when **P** is increased to 8 is expected due to the additional time for managing the process pool.

5.7 Chapter Summary

In this chapter, we present an empirical analysis showing that different topics have different likelihood of getting propagated at both network and individual levels. We then propose to model the virality and susceptibility factors at topic level. We develop **V2S**, a tensor factorization based framework, and its associated models to learn topic-specific user virality, topic-specific user susceptibility, and topic virality from content propagation data. Our experiments on a large Twitter dataset have shown that the proposed **V2S**-based models outperform baseline models significantly in propagation prediction. Our experiments on synthetic databases have also shown that our proposed models outperform all the other baseline methods in learning the topic-specific factors. Part of the work in this chapter has been published in [85].

Part II

Modeling Community Behavior

Chapter 6

Modeling of Community Behaviors and Sentiments

In this chapter, we propose a model for determining the community affiliation of microblogging users based on their content, the sentiments they express on their content, and the behaviors they adopt. This chapter is organized as follows. We first discuss the research task in Section 6.1. We also state our objectives and summarize our contributions in this section. We then present our proposed model in Sections 6.2. Next, we describe two experimental datasets in Section 6.3. The experimental evaluation of the proposed model on the two datasets is reported in Sections 6.4 and 6.5 respectively. Finally, we conclude the chapter in Section 6.6.

6.1 Introduction

Recent empirical works have shown some strong correlations between a microblogging user's community affiliation and the topic and sentiment expressed in her tweets, as well as her behaviors [99, 60, 203, 83, 228]. Previous studies have attempted to model user communities based on one or some of the content, sentiment, and behaviors factors [169, 29, 248]. However, to the best of our knowledge, there are no works that considers all these factors in modeling user communities.

In this chapter, we postulate that, other than user content, sentiments expressed on the content's topics and other microblogging behaviors of a user can be shaped by her community affiliations. For example, users belonging to a political community may be more interested in retweeting each other, or express positive sentiment on issues they support but negative sentiment on those they oppose. We therefore aim to develop a new model that simultaneously derives the community of each user, and the common behaviors and common topic-specific sentiment of each community. This research task is however challenging due to the following reasons:

- Multiple types of user behaviors have to be treated differently, but modeled in a unified way.
- Topic and sentiment of tweets are not known before hand. One either has to first determine the topics and sentiments before using them in modeling user behaviors and communities, or to learn them as part of the model.

This chapter addresses the first challenge by developing a general framework that allows user content as well as user behaviors of different types to be modeled as different "bag-of-words". We address the second challenge by coupling with an existing sentiment analysis tool for microblogging. Lastly, we develop a probabilistic graphical model that simultaneously infers latent topics, users' topic interests, latent communities and their associated behaviors and topic-specific sentiments.

Our main contributions in this chapter consist of the following.

- We propose a probabilistic graphical model, called **CBS**, for mining topics and user communities, as well as mining behaviors and topic-specific sentiments associated with the communities.
- We develop a sampling method to infer the model's parameters.

- We apply **CBS** model on two real politics related Twitter datasets and show that it outperforms other baseline topic models.

- An empirical analysis of behaviors and topic-specific sentiments for the two datasets has been conducted to demonstrate the efficacy of the **CBS** model.

6.2 The CBS Model

In this section, we present our proposed model in detail. We first introduce notations and assumptions used in this chapter. Next, we describe the model and the sampling method for learning the model's parameters.

6.2.1 Notations

We summarize the notations used to describe the **CBS** model in Table 6.1. Consider a dataset of Twitter users together with their posted tweets and behavior traces, we use U and L to denote the number of users and the number of behavior types in the dataset respectively. For each user u, we denote the set of M_u tweets she posts by $\boldsymbol{\mathcal{T}}_u = \{t_u^1, \cdots, t_u^{M_u}\}$; and denote the set of all the tweets in the dataset by $\boldsymbol{\mathcal{T}}$, i.e., $\boldsymbol{\mathcal{T}} = \bigcup\limits_u \boldsymbol{\mathcal{T}}_u$. Each tweet t_u^j is a bag-of-words with length N_u^j, i.e., $t_u^j = \{w_u^{j1}, \cdots, w_u^{jN_u^j}\}$, where each word w_u^{jn} is drawn from a common vocabulary of W_t words $\boldsymbol{\mathcal{V}}_t = \{w_1, \cdots, w_W\}$. Also, for each tweet t_u^j, we denote its topic and sentiment by z_u^j and s_u^j respectively. The bag-of-topics and the bag-of-sentiments of all the tweets is denoted by $\boldsymbol{\mathcal{Z}}$ and $\boldsymbol{\mathcal{S}}$ respectively.

Similarly, for each user u, and each behavior type l, we use $\boldsymbol{\mathcal{B}}_u^l$ to denote the length-B_u^l bag-of-behaviors of type l that u adopts, i.e., $\boldsymbol{\mathcal{B}}_u^l = \{b_u^{l1}, \cdots, b_u^{lB_u^l}\}$. Each behavior b_u^{lj} is drawn from a common behavior type-l vocabulary $\boldsymbol{\mathcal{V}}_b^l$. We denote the number of behaviors in the vocabulary by W_b^l, i.e., $W_b^l = |\boldsymbol{\mathcal{V}}_b^l|$. Lastly, we use $\boldsymbol{\mathcal{B}}$ to denote the bag-of-all-behaviors (of all types) of all the users.

Table 6.1: Notations used to describe **CBS** model

U/ L	Number of users/ Number of user behavior types
$\boldsymbol{\mathcal{V}}_t$	Tweet vocabulary
W_t	Number of words in tweet vocabulary, $W = \lvert\boldsymbol{\mathcal{V}}\rvert$
$\boldsymbol{\mathcal{T}}$	Set of all tweets
$\boldsymbol{\mathcal{T}}_u$	Set of tweets posted by user u
t_u^j	j-th tweet of user u
$\boldsymbol{\mathcal{T}}_{-t_u^j}$	Set of all tweets except t_u^j
M_u	Number of tweets posted by user u, $M_u = \lvert\boldsymbol{\mathcal{T}}_u\rvert$
w_u^{jn}	n-th word in tweet t_u^j
N_u^j	Number of words in tweet t_u^j
$\boldsymbol{\mathcal{V}}_b^l$	Behavior type-l vocabulary
W_b^l	Number of behaviors of type-l, $W_b^l = \lvert\boldsymbol{\mathcal{V}}_b^l\rvert$
$\boldsymbol{\mathcal{B}}$	Bag-of-all-behaviors of all types
$\boldsymbol{\mathcal{B}}_u^l$	Bag-of-behaviors of type l of user u
b_u^{lj}	j-th behavior of type-l of user u
B_u^l	Number of behaviors of type-l of user u, $B_u^l = \lvert\boldsymbol{\mathcal{B}}_u^l\rvert$
C	Number of communities
π	Community distribution
c_u	Community of user u
$\boldsymbol{\mathcal{C}}$	Bag-of-communities of all users
$\boldsymbol{\mathcal{C}}_{-u}$	Bag-of-communities of all users except u
δ_{ck}	Topic-specific sentiment distribution of community c for topic k
λ_{cl}	Behavior distribution of community c for type-l behaviors
K	Number of topics
θ_u	Topic distribution of user u
ϕ_k	Word distribution of topic k
z_u^j/ s_u^j	Topic/ sentiment of tweet t_u^j
$\boldsymbol{\mathcal{Z}}$/ $\boldsymbol{\mathcal{S}}$	Bag-of-topics/ bag-of-sentiments of all tweets
$\boldsymbol{\mathcal{Z}}_{-t_u^j}$/ $\boldsymbol{\mathcal{S}}_{-t_u^j}$	Bag-of-topics/ bag-of-sentiments of all tweets except t_u^j
τ/ α/ η/ γ_l	Dirichlet prior of π/ θ_u/ σ_{ck}/ λ_{cl}
$\mathbf{n_c}(c, \boldsymbol{\mathcal{C}})$	Number of times community c is observed in bag-of-communities $\boldsymbol{\mathcal{C}}$
$\mathbf{n_s}(s, z, c, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{Z}})$	Number of times sentiment s is observed in topic z in the set of tweets posted by users of community c for bag-of-sentiments $\boldsymbol{\mathcal{S}}$ and bag-of-topics $\boldsymbol{\mathcal{Z}}$
$\mathbf{n_b}(b, c, \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{C}})$	Number of times behavior b is adopted by users of community c for the bag-of-behaviors $\boldsymbol{\mathcal{B}}$ and the bag-of-communities $\boldsymbol{\mathcal{C}}$
$\mathbf{n_w}(w, z, \boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{Z}})$	Number of times word w is observed in the topic z for the set of tweets $\boldsymbol{\mathcal{T}}$ and the bag-of-topics $\boldsymbol{\mathcal{Z}}$
$\mathbf{n_z}(z, u, \boldsymbol{\mathcal{Z}})$	Number of times topic z is observed in the set of tweets posted by user u for the bag-of-topics $\boldsymbol{\mathcal{Z}}$.

6.2.2 Assumptions

The basic assumption of our model is that while users within a community may have different topical interest in tweeting, they should adopt similar behaviors. We therefore assume that, for *each type of behaviors*, each community has a certain interest in some behaviors of the type, and all the users within the community adopt the behaviors following this interest. For example, a Christian often mentions religion in her biography, or a football fan often follows and retweets from her supporting team's pages. Moreover, different communities may express different sentiments on the same topic, e.g., Democrats are more positive about healthcare issues while Republicans are more negative. Hence, behaviors a user adopted and sentiment she expressed in her tweets are useful in identifying the community that she belongs to.

6.2.3 Generative Process

The **CBS** model has K latent topics, where each topic k has a multinomial distribution ϕ_k over the vocabulary $\boldsymbol{\mathcal{V}}_t$. As tweets are short with no more than 140 characters, we assume that each tweet has only one topic. Each user u belongs to one of C communities, following the (global) community distribution π. Each user u has a topic distribution θ_u, while each community c has a topic-specific sentiment distribution σ_{ck} for each topic k. Moreover, for each behavior type l, each community c has a multinomial distribution λ_{cl} over the set of all type-l behaviors. Lastly, we assume that π, θ_u, σ_{ck}, and λ_{cl} have Dirichlet priors τ, α, η, and γ_l respectively.

In summary, the **CBS** model has the plate notation as shown in Figure 6.1 and the generative process as follows.

- Sample the community distribution vector $\pi \sim Dirichlet(\tau)$
- For each $k = 1, \cdots, K$, sample the k-th topic $\phi_k \sim Dirichlet(\beta_k)$
- For each community c and each topic k, sample the topic-specific senti-

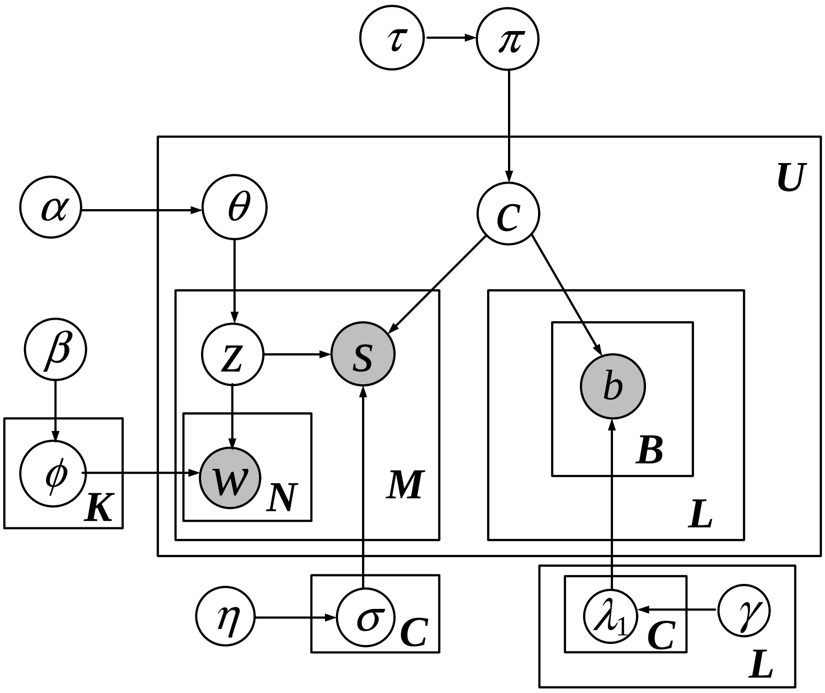


Figure 6.1: Plate notation for **CBS** model

ment distribution $\sigma_{ck} \sim Dirichlet(\eta_{ck})$

- For each community c, and each type of behavior l, sample type-l behavior distribution $\lambda_{cl} \sim Dirichlet(\gamma_{cl})$
- For each user u, sample community indicator $c_u \sim Multinomial(\pi)$
- For user u, generate tweets for the user:

  1. Sample topic distribution $\theta_u \sim Dirichlet(\alpha)$
  2. For each tweet t:

     (a) Sample topic for the tweet $z_t \sim Multinomial(\theta_u)$

     (b) Sample tweet's words: for each word slot n, sample the word $w_{t,n} \sim Multinomial(\phi_{z_t})$

     (c) Sample sentiment for the tweet $s_t \sim Multinomial(\sigma_{cz_t})$

- Generate behaviors for each user u:

  For each behavior of type $l = 1, \cdots, L$, and for each $n = 1, \cdots, B_{ul}$, sample the behavior $b \sim Multinomial(\lambda_{cl})$

Note that in **CBS** model, we currently determine the sentiments of tweets using Stanford's sentiment scoring API[1,2]. The widely used Stanford's sentiment scoring API implements a machine learning method to detect sentiment expressed in a tweet purely based on content of the tweet. For each tweet, the API returns a score of 4, 0, or 2 to indicate the tweet is positive, negative, or neutral respectively.

6.2.4 Model Learning

Due to the intractability of LDA-based model [27], we make use of sampling method in learning and estimating the parameters in the model. More exactly, we use a collapsed Gibbs sampler to iteratively sample the latent community of every user, and latent topic of every tweet.

Assume that the current user we have to sample the community for is u. We use $\boldsymbol{\mathcal{C}}_{-u}$ to denote the bag-of-communities of all other users in the dataset except u. Similarly, for each tweet t_j^u, we use $\boldsymbol{\mathcal{Z}}_{-t_u^j}$ and $\boldsymbol{\mathcal{S}}_{-t_u^j}$ to denote the bag-of-topics and bag-of-sentiments, respectively, of all other tweets in the dataset except t_u^j. Finally, for each behavior b_u^{lj}, we use $\boldsymbol{\mathcal{B}}_{-b_u^{lj}}$ to denote the bag-of-behaviors excluding b_u^{lj}. Then, the community of u is sampled according to Equation 6.1.

$$p(c_u = c|\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{C}}_{-u}, \boldsymbol{\mathcal{Z}}, \alpha, \beta, \tau, \eta, \lambda) \propto \prod_{j=1}^{M_u} \frac{\mathbf{n_s}(s_u^j, z_u^j, c, \boldsymbol{\mathcal{S}}_{-s_u^j}, \boldsymbol{\mathcal{Z}}_{-z_u^j}) + \eta_{cz_u^j s_u^j}}{\sum_{q=1}^{C}(\mathbf{n_s}(s_u^j, z_u^j, q, \boldsymbol{\mathcal{S}}_{-s_u^j}, \boldsymbol{\mathcal{Z}}_{-z_u^j}) + \eta_{qz_u^j s_u^j})} \cdot \prod_{l=1}^{L}\prod_{j=1}^{B_u^l} \frac{\mathbf{n_b}(b_u^{lj}, c, \boldsymbol{\mathcal{B}}_{-b_u^{lj}}, \boldsymbol{\mathcal{C}}_{-u}) + \lambda^l_{cb_u^{lj}}}{\sum_{b=1}^{W_b^l} \mathbf{n_b}(b, c, \boldsymbol{\mathcal{B}}_{-b_u^{lj}}, \boldsymbol{\mathcal{C}}_{-u}) + \lambda^l_{cb}} \cdot \frac{\mathbf{n_c}(c, \boldsymbol{\mathcal{C}}_{-u}) + \tau_c}{\sum_{q=1}^{C}(\mathbf{n_c}(q, \boldsymbol{\mathcal{C}}_{-u}) + \tau_q)} \tag{6.1}$$

Now, we have to sample the topic for the current tweet denoted by t_u^j. Let $\boldsymbol{\mathcal{T}}_{-t_u^j}$ denotes the set of all tweets in the dataset excluding t_u^j. Then topic of t_u^j

[1] http://help.sentiment140.com/api

[2] This also reduces the complexity of **CBS** model as sentiment mining itself is already well studied research problem.

is sampled according to Equation 6.2.

$$
\begin{aligned}
p(z_u^j = z | \boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{Z}}_{-t_u^j}, \alpha, \beta, \tau, \eta\lambda, \eta) \propto \prod_{n=1}^{N_u^j} \frac{\mathbf{n_w}(w_u^{jn}, z, \boldsymbol{\mathcal{T}}_{-t_u^j}, \boldsymbol{\mathcal{Z}}_{-t_u^j}) + \beta_{zw_u^{jn}}}{\sum_{v=1}^{W_t} (\mathbf{n_w}(v, z, \boldsymbol{\mathcal{T}}_{-t_u^j}, \boldsymbol{\mathcal{Z}}_{-t_u^j}) + \beta_{zv})} \cdot \\
\cdot \frac{\mathbf{n_s}(s_u^j, z, c_u, \boldsymbol{\mathcal{S}}_{-t_u^j}, \boldsymbol{\mathcal{Z}}_{-t_u^j}, \boldsymbol{\mathcal{C}}) + \eta_{c_u z s_u^j}}{\sum_{p=1}^{P} (\mathbf{n_s}(p, z, c_u, \boldsymbol{\mathcal{S}}_{-_u^j}, \boldsymbol{\mathcal{Z}}_{-t_u^j}, \boldsymbol{\mathcal{C}}) + \eta_{c_u z p})} \cdot \frac{\mathbf{n_z}(z, u, \boldsymbol{\mathcal{Z}}_{-t_u^j}) + \alpha_z}{\sum_{k=1}^{K} (\mathbf{n_z}(k, u, \boldsymbol{\mathcal{Z}}_{-t_u^j}) + \alpha_k)} \qquad (6.2)
\end{aligned}
$$

In Equations 6.1 and 6.2, $\mathbf{n_c}(c, \boldsymbol{\mathcal{C}})$ records the number of times the community c observed in the bag-of-communities $\boldsymbol{\mathcal{C}}$. Similarly, $\mathbf{n_s}(s, z, c, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{Z}})$ records the number of times the sentiment s observed in the topic z in the set of tweets posted by users of community c for bag-of-sentiments $\boldsymbol{\mathcal{S}}$ and bag-of-topics $\boldsymbol{\mathcal{Z}}$. Next, $\mathbf{n_b}(b, c, \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{C}})$ records the number of times the behavior b is adopted by users of community c for the bag-of-behaviors $\boldsymbol{\mathcal{B}}$ and the bag-of-communities $\boldsymbol{\mathcal{C}}$; and $\mathbf{n_w}(w, z, \boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{Z}})$ records the number of times the word w is observed in the topic z for the set of tweets $\boldsymbol{\mathcal{T}}$ and the bag-of-topics $\boldsymbol{\mathcal{Z}}$. Lastly $\mathbf{n_z}(z, u, \boldsymbol{\mathcal{Z}})$ records the number of times the topic z is observed in the set of tweets posted by user u for the bag-of-topics $\boldsymbol{\mathcal{Z}}$.

In our experiments, we used symmetric Dirichlet hyperparameters with $\alpha = 50/K$, $\beta = 0.01$, $\tau = 5$, $\eta = 5$, and $\gamma_l = 0.01$ for all $l = 1, \cdots, L$. Each time, we run the model for 300 iterations of Gibbs sampling. We take 20 samples with a gap of 5 iterations in the last 100 iterations to assign values to all the hidden variables.

6.3 Datasets

In order to get clear notions of communities and topics, the following two politically oriented datasets were used for evaluating the **CBS** model.

MoC dataset. The first dataset consists of tweets posted by members of the 112th U.S congress. We manually identified the official Twitter accounts of 93 senators (47 Democrats and 46 Republicans) and collected their tweets in the duration of May 2012 - Feb 2013. In other words, we have the ground truth political affiliations of all users in this dataset.

One-Week dataset. The second dataset is large set of tweets generated just before the 2012 US presidential election. We first manually selected 56 *seed users* who are popular political related figures with many followers on Twitter. These include major American politicians, such as 2012 US presidential candidates, e.g., Barack Obama, Mitt Romney, and Newt Gingrich; well known political bloggers in U.S., e.g., America Blog, Red State, and Daily Kos; and political sections of US news media, e.g., CNN Politics, and Huffington Post Politics. The set of users were then expanded by adding all users following at least three seed users. This resulted in 23,992 users whose biographies are collected. Based on their biographies, we were able to manually label the political affiliations of 2,319 of them, including 202 Democrats, 228 Neutrals and 1709 Republicans. The following links of these users were then collected. Since users in this dataset have different degree of political involvement, their tweets cover not only politics but also a variety of other topics. To focus on political topics, we extracted only the political tweets from all tweets posted in the first week of October 2012 using a keyword-based filter. The keywords are political hashtags and political topics' representative words/phrases identified by the semi-automatic method presented in [83].

Data preprocessing. We employed the following preprocessing steps to clean both the datasets. We first removed all stopwords from the tweets. Then, for **MoC** dataset, we removed all tweets containing stopwords only and users with less than 5 (remaining) tweets. For **One-Week** dataset, we removed all tweets with less than 3 non-stopwords and and users with less than 10 tweets. In **MoC** dataset, we consider the following behavior types for each user: (1) *user mention*, and (2) *hashtag*; while in **One-Week** dataset, behavior types a user may perform are: (1) *user mention*, and (2) *hashtag*, (3) *retweet*, (4) *followee*, and (5) *profile word* (i.e., non-stopwords in the user's biography). The

Table 6.2: Statistics of the experimental datasets used for evaluating **CBS** model

			MoC *dataset*	**One-Week** dataset
#user	Total		93	23,992
	With political label	All labels	93	2,193
		Democrat	47	202
		Neutral	0	228
		Republican	46	1,709
#tweets			87,182	839,687
#behaviors	mention		14,609	68,804
	hashtag		26,152	561,098
	retweet		-	181,661
	followee		-	24,044,367
	profile word		-	64,107

hashtag, *retweet*, and *user mention* behaviors are further divided into positive, neutral, or negative depending on whether the behavior is contained in a positive, neutral, or negative tweet. For each of those behavior, we assign a *_(+)*, *_(0)*, or *_(-)*) suffix to indicate that if the behavior is positive, neutral, or negative respectively. For example, if user u mentions *BarrackObama* in a positive tweet (respectively neutral and negative), then we have *BarakObama_(+)* (respectively *BarakObama_(0)* and *BarakObama_(-)*) in the bag-of-user-mentions of the user. Lastly, for each behavior, for **MoC** we filtered out all the behaviors with less than 5 users performing the behavior, while for **One-Week** dataset we filtered out all the behaviors with less than 50 users performing the behavior.

The reasons that, in the preprocessing steps, we used higher thresholds for **One-Week** dataset than for **MoC** dataset are: (1) we expected that the former contains much more noise than the latter, and (2) the former has a much larger number of users than the latter, and we wanted to focus on global behaviors rather than local behaviors. Table 6.2 shows the statistics of the two datasets after the preprocessing steps.

6.4 Experiments on MoC dataset

In this experiment, we evaluate the performance of **CBS** model and other baseline methods in topic modeling and user clustering tasks using **MoC** dataset.

Topic modeling task. Proposed by Zhao *et. al.* [258], TwitterLDA is a variant of LDA [27], a commonly used method for topic modeling. TwitterLDA constrains each tweet to have only one topic. This constraint is appropriate for short documents as well as tweets. We will therefore compare **CBS** and TwitterLDA based on their abilities to model topics as the number of topics is varied from 10 to 100. We expect that the two models have similar performance in this task as they share the same way to model the topics of the tweets.

User clustering task. To evaluate the performance of **CBS** in user clustering, we compare it with K-means clustering. To implement K-means clustering, we represent each user as a vector of features, where the features include (1) topic distribution of tweets posted by the user, and (2) bags-of-behaviors of the users. The topic distribution of tweets posted by a user is discovered using TwitterLDA model with the number of topics is set to 70 as will be explained below. In this task we expect that **CBS** outperforms K-mean as the former uses more information to cluster the users than the latter. K-mean uses users' topics and behaviors only, while **CBS** also use the sentiments users express on different topics.

6.4.1 Evaluation Metrics

We adopt *likelihood* and *perplexity* for evaluating the topic modeling task. For each user, we randomly selected 90% of tweets of the user to form a training tweet set, and use the remaining 10% of the tweets as the test tweet set. Then for each method, we compute the likelihood of the training tweet set and perplexity of the test tweet set. The method with a higher likelihood, or lower perplexity is considered better for the task.

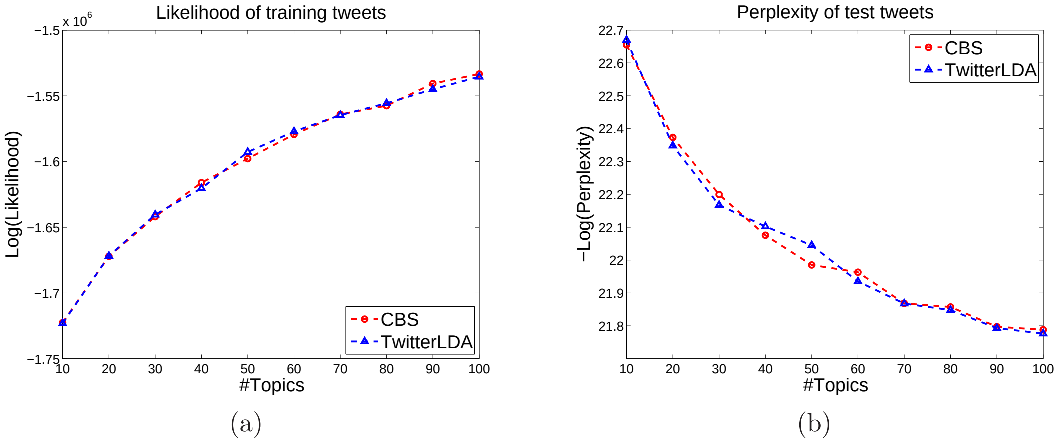


Figure 6.2: **MoC** dataset: performance of different models in topic modeling

For user clustering task, we adopt *weighted entropy* as the performance metric. As we have two different political affiliations in the dataset, we run the methods with the number of communities set to 2. We finally computed the weighted entropy of the resultant communities as follows.

$$E = -\sum_{c=0}^{1} \frac{n_c}{U} * \left[\frac{n_c^D}{n_c} * log\frac{n_c^D}{n_c} + \frac{n_c^R}{n_c} * log\frac{n_c^R}{n_c}\right] \tag{6.3}$$

where n_c is the number of users assigned to community c, and n_c^D and n_c^R are the numbers of Democrats and Republicans assigned to community c respectively. Recall that $U = 93$ is the number of users in the dataset. The method with a lower entropy is the winner in the task.

6.4.2 Performance Results

Figure 6.2 shows the performance of TwitterLDA and **CBS** model in topic modeling. As expected, (1) the two models yield very similar likelihood and perplexity; and (2) larger number of topics K gives larger likelihood and smaller perplexity, and the amount of improvement diminishes as K increases. Considering both time and space complexities, we set the number of topics to be 70 for the user clustering task.

Figure 6.3 shows the performance of K-mean and **CBS** models in user

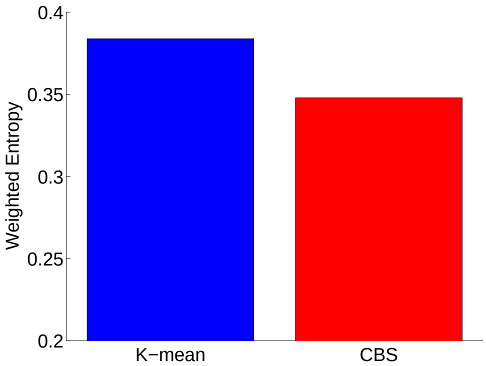


Figure 6.3: **MoC** dataset: performance of different models in user clustering

clustering. Again, as expected, the figure clearly shows that **CBS** outperforms K-means in user clustering. **CBS** therefore is a better solution for user clustering than the combination of TwitterLDA and K-means.

6.4.3 Topic Sentiment Analysis

We now analyze the topic sentiment results of **CBS** model on **MoC** dataset. For the two learnt communities, we assign each community to be Democrat or Republican if most users in the community are democrat or republican respectively.

Table 6.3 shows the top positive topics and top negative topics of each community as obtained by **CBS**. Note that the topic labels are manually assigned based on examining the topics' top words and top tweets. For each topic, the topic's top words are the words having the highest likelihoods given the topic, and the topic's top tweets are the tweets having the lowest perplexities given the topic. Table 6.3 shows that those extreme topics are reasonable. On one hand, the two communities share the common sentiment on topic about broadcasting the talks/shows by senators of the same party (Topic 16, Topic 23), or nationwide common topics like greetings for vacation and holidays (Topic 16), victories of U.S. team in Olympic 2012 (Topic 29), shooting and terrorism (Topic 34). On the other hand, the two communities are negative on different topics: the Democrat community is negative on topics on legislative issues and

Table 6.3: **MoC** dataset: top positive and negative topics per community

		Topic ID	Topic Label
Democrat	Positive	Topic 16	Greetings
		Topic 29	U.S. teams in Olympic 2012
		Topic 44	Live talks
	Negative	Topic 34	Shooting & terroism
		Topic 32	Legislative issues
		Topic 26	Economics issues
Republican	Positive	Topic 23	Live shows
		Topic 16	Greetings
		Topic 29	US teams in Olympic 2012
	Negative	Topic 34	Shooting & terroism
		Topic 25	Financial issues
		Topic 43	Recovering from Sandy hurrican

economics issues, which mostly under control of Republicans, while the Republican community is negative on the process of recovering from Sandy hurricane (Topic 43) and financial issues, which are mostly raised by Democrats.

6.4.4 Behavior Analysis

Next, we look into the community representative behaviors uncovered by **CBS** from the **MoC** dataset. Table 6.4 shows the top hashtags and top user mentions by users in each community. The table clearly shows that those extreme behaviors are also reasonable. For *hashtag*, all the top hashtags are neutral, and the top ones of each community are most popular hashtags among Twitter users of the community. For *user mention*, the top mentioned users in the Democrat community are democrat users (e.g., *BarackObama*), goverment officers (e.g., *speakerboehner*), or pro-democrat media (e.g., *msnbc*), while the top mentioned user in the Republican community are republican senators (e.g., *johncornyn*), and pro-republican media (e.g., *foxnews*).

6.5 Experiments on One-Week dataset

In this section, we report our experiments on **One-Week** dataset. Given the large number of users and tweets, and a partial ground truth of users' political affiliations in the dataset, we evaluate **CBS** and other comparative methods

Table 6.4: MoC dataset: top behaviors per community

Hashtag		**User mention**	
Democrat	Repulican	Democrat	Repulican
#jobs_(0)	#tco_(0)	@speakerboehner_(0)	@wsj_(0)
#nj_(0)	#tcot_(0)	@barackobama_(0)	@foxnews_(+)
#vawa_(0)	#obamacare_(0)	@whitehouse_(0)	@foxnews_(0)
#senate_(0)	#sayfie_(0)	@fema_(0)	@johncornyn_(+)
#sandy_(0)	#fiscalcliff_(0)	@msnbc_(+)	@grahamblog_(0)
#veterans_(0)	#jobs_(0)	@markudall_(0)	@johncornyn_(0)
#budget_(0)	#libya_(0)	@senatorcollins_(0)	@mittromney_(+)
#gop_(0)	#gop_(0)	@nytimes_(0)	@senate_(0)
#job_(0)	#syria_(0)	@senatormenendez_(0)	@senatorayotte_(0)
#socialsecurity_(0)	#debt_(0)	@barackobama_(+)	@joelieberman_(0)

in topic modeling and user classification tasks.

Topic modeling task. Similar to the experiments presented in Section 6.4, we compare **CBS** with TwitterLDA based on their abilities to model topics as the number of topics is varied from 10 to 100.

User classification task. We formulate the user classification task as a semi-supervised learning problem since: (1) we have ground truth of political affiliations for only 10% of the users in the dataset, and (2), as shown in [45], the supervised learning approach for users' political affiliation classification in microblogging is not practical given the users having different degree of political involvement like in **One-Week** dataset. To evaluate the performance of **CBS** in this task, we therefore compare it with semi-supervided learning (SSL) methods provided in Junto toolbox[3], which are shown to be among state-of-the-art semi-supervised learning methods[207]. The Junto toolbox implements label propagation methods which iteratively update label for each (unknown label) user u based on labels of the other users who are most similar to u. Here, we choose to use the cosine similarity between pairs of users. To do this, we represent each user as a vector of features, where the features are: (a) tweet-based features, and (b) bags-of-behaviors of the users. We employ two ways to compute tweet-based features for each user: (1) Tf-Idf based: the features of each user are TF-IDF scores [150] of the terms contained in the user's

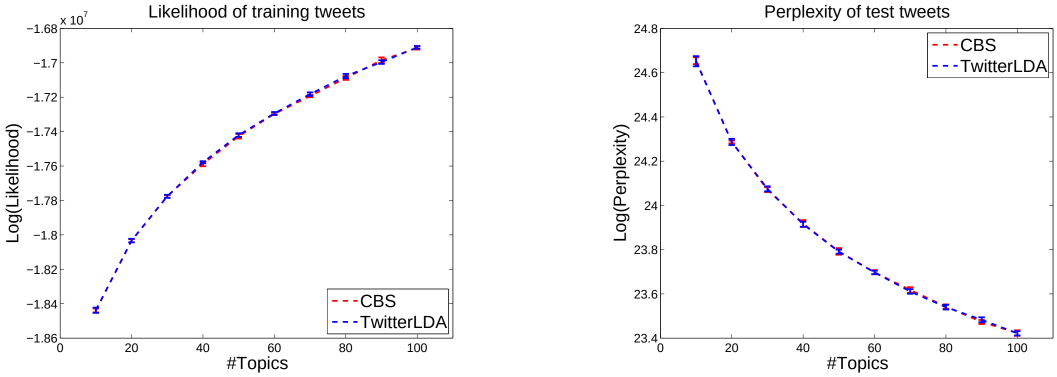
[3] https://github.com/parthatalukdar/junto

tweets; and (2) TwitterLDA based: the features of each user are the components in topic distribution of the user's tweets discovered by TwitterLDA model. For computing the TwitterLDA based features, we set the number of topics in TwittterLDA model to 80 as will be explained below. Again, we expect that **CBS** outperforms the SSL methods in this task as the former uses more information to classify the users than the latter.

6.5.1 Evaluation Metrics

Again, we adopt *likelihood* and *perplexity* for evaluating the topic modeling task. Similarly to the experiment in Section6.4, for each user, we randomly selected 90% of tweets of the user to form training tweets set, and use the remaining 10% of the tweets as the test tweets set. Then for each method, we computed the likelihood of the training tweets set and perplexity of the test tweets set. Method with a higher likelihood, or lower perplexity is considered better for the task.

For user classification task, we adopt *average $F1$ score* as the performance metric. To do this, we first evenly distributed the set of known political affiliation users in 10 folds such that the folds have the same fraction of Democrat/Neutral/Republican users. Then, for each method, we run 10-fold cross validation with number of communities set to 3 (corresponding to three different political affiliations in the dataset). More precisely, for each method and each time, we use 9 folds of known political affiliation users and all unknown political affiliation users as (semi-)training set, and use the remaining fold of known political affiliation users as test set. For **CBS** model, in the training phase, we set Democrat, Neutral, and Republican to be community 0, 1, and 2 respectively. We also fix the community indicators of the users in the 9 folds of the (semi-)training set according to their ground truth political affiliation (i.e., we do not sample community for those users). We then compute the average $F1$ score obtained by each method in all three classes (i.e., Democrat, Neutral,



(a) (b)

Figure 6.4: **One-Week** dataset: performance of different models in topic modeling

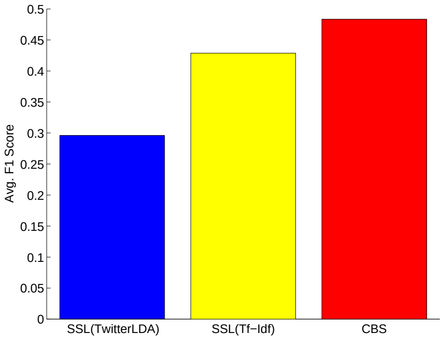


Figure 6.5: **One-Week** dataset: performance of different models in user classification

and Republican). The method with a higher score is the winner in the task.

6.5.2 Performance Results

Figure 6.4 shows the performance of TwitterLDA and **CBS** model in topic modeling. The likelihood and perplexity values in the figure are averaged over 10 runs. Again, as we expected, more topics K gives larger likelihood and smaller perplexity, and the amount of improvement diminishes as K increases. Similar to what reported in Section 6.4, the figure shows that the topic modeling performance of **CBS** and TwitterLDA are very similar. This suggests that **CBS** model is robust against the changes in the bag-of-behaviors used.

Based on Figure 6.4 results, and in consideration of both time and space complexities, we set the number of topics to 80 for the user classification task.

The performance of **CBS** and the SSL methods in user classification task is shown in Figure 6.5. The SSL(TwitterLDA) (respectively SSL(Tf-Idf)) is the best performance obtained by methods provided in the Junto toolbox where the users' tweet-based features are TwitterLDA-based features (respectively Tf-Idf based features). The fact that SSL(Tf-Idf) outperforms SSL(TwitterLDA) can be explained that TwitterLDA suffers from noise as, within only one week, many users do not have many tweets for their topic distribution to be inferred correctly by TwitterLDA model. Finally, as we expected, the figure clearly shows that our **CBS** model is the best among all the methods.

6.5.3 Topic Sentiment Analysis

We now analyze the results obtained from applying **CBS** model on **One-Week** dataset. Table 6.5 shows the top positive topics and top negative topics of each community as obtained by **CBS**. Again, we have manually assigned labels for those topics by examining the topics' top words and top tweets. The table shows that those extreme topics are reasonable. In one end, while the two wings, i.e., the Democrat and the Republican communities, are positive on the election related topics, e.g., calling for vote for the one building the nation (Topic 58), or tweeting about politics using sport terms (Topic 3), the Neutral is more positive in tweeting about protecting the country (Topic 60), and changes in economics (Topic 62). Also, it is expected that both the Democrat and the Neutral community are positive on Mr&Mrs Obama's anniversary (Topic 57). On the other end, while all three communities are negative on financial issues (Topic 20) and military issues (Topic 66), the Democrat community is more negative on issues raised by the conservatives (Topic 26), but the Neutral and the Republican communities are more negative on the tax policy (Topic 21).

Table 6.5: **One-Week** dataset: top positive and most negative topics per community

		Topic	Topic Label
Democrat	Positive	Topic 57	Mr&Mrs Obama's anniversary
		Topic 58	Voting for national building
		Topic 3	Politics as a sport game
	Negative	Topic 26	Conservative issues
		Topic 66	Military issues
		Topic 20	Financial issues
Neutral	Positive	Topic 57	Mr&Mrs Obama's anniversary
		Topic 60	Protecting the country
		Topic 62	Economics changes
	Negative	Topic 20	Financial issues
		Topic 66	Military Policy
		Topic 21	Tax policy
Republican	Positive	Topic 3	Politics as a sport game
		Topic 47	Campaining
		Topic 58	Voting for national building
	Negative	Topic 20	Financial issues
		Topic 21	Tax policy
		Topic 66	Military Policy

6.5.4 Behavior Analysis

Table 6.6 shows the top behaviors performed by users in each community of all five behavior types. The table clearly shows that those extreme behaviors are also reasonable. The top profile words of each community are representative ones for the community: *liberal, progressive, democrats*, etc. for the Democrat community; *conservative, christian, #tcot*, etc. for the Republican community; and *media, sport, music, editor*, etc. for the Neutral community, which including most of accounts of professional persons/associations. For *followee* behavior, it is expected that the top followed users of the Democrat and the Republican communities are most popular ones in each community respectively, while the top ones of the Neutral community are mostly goverment office (e.g. *WhiteHouse*) and media (e.g., *nytimes*, *BreakingNews*, and *AP*). Similarly, for *retweet*, the top retweeted users of Democrat and Republican communities are most popular ones in each community respectively, while the top ones of Neutral community are mostly media. The top hashtags suggest that the two wings (i.e., the Democrat and the Republican communities) tweet most about topics within their own community (e.g.,*#p2* for the Democrat community,

Table 6.6: **One-Week** dataset: top behaviors per community

Profile			**Followee**		
Democrat	Neutral	Repulican	Democrat	Neutral	Repulican
liberal	politics	conservative	BarackObama	BarackObama	michellemalkin
love	media	love	maddow	WhiteHouse	PAC43
politics	love	christian	thinkprogress	politico	KatyinIndy
progressive	sports	god	WhiteHouse	nytimes	BraveLad
lover	music	american	MotherJones	mittromney	Heritage
obama	world	country	TheDailyEdge	BreakingNews	Miller51550
democrat	editor	#tcot	DavidCornDC	WSJ	SarahPalinUSA
mom	student	wife	billmaher	cnnbrk	AndyWendt
music	tweets	family	dccc	AP	seanhannity

Retweet		
Democrat	Neutral	Repulican
thedailyedge_(0)	thinkprogress_(0)	patdollard_(0)
barackobama_(0)	reuters_(0)	jjauthor_(0)
thinkprogress_(0)	barackobama_(0)	newsninja2012_(0)
lolgop_(0)	ap_(0)	mittromney_(0)
thenewdeal_(0)	drudge_report_(0)	slone_(0)
truthteam2012_(0)	thedailyedge_(0)	katyinindy_(0)
jeffersonobama_(0)	truthteam2012_(0)	connewsnow_(0)
chrisrockoz_(0)	huffpostpol_(0)	iowahawkblog_(0)
bluedupage_(0)	patdollard_(0)	keder_(0)

Hashtag			**User mention**		
Democrat	Neutral	Repulican	Democrat	Neutral	Repulican
#p2_(0)	#tcot_(0)	#tcot_(0)	@mittromney_(0)	@barackobama_(0)	@barackobama_(0)
#romney_(0)	#obama_(0)	#obama_(0)	@cspanwj_(0)	@mittromney_(0)	@mittromney_(0)
#gop_(0)	#syria_(0)	#teaparty_(0)	@mittromney_(+)	@mittromney_(+)	@mittromney_(+)
#tcot_(0)	#p2_(0)	#p2_(0)	@barackobama_(0)	@barackobama_(+)	@barackobama_(+)
#obama_(0)	#iran_(0)	#tlot_(0)	@barackobama_(+)	@cnn_(0)	@youtub_(0)
#tco_(0)	#romney_(0)	#gop_(0)	@cspanwj_(+)	@barackobama_(-)	@breitbartnew_(0)
#obama2012_(0)	#gop_(0)	#tco_(0)	@thinkprogres_(0)	@mittromney_(-)	@sharethi_(0)
#p_(0)	#news_(0)	#romney_(0)	@edshow_(0)	@abc_(0)	@cnn_(0)
#p2b_(0)	#tco_(0)	#lnyhbt_(0)	@maddow_(0)	@paulryanvp_(0)	@seanhannity_(0)

and *#tcot* for the Republican community) and then about the opposite one, while the Neutral community tweets more about topics related to international issues (e.g., *#syria*, *#iran*). For *user mention* behavior, it is interesting that while the Neutral community mentions the two candidate equally, users of the two wings mention the opposite candidate more. This due to the fact that, during the campaign period, the wing users often mention the opposite candidate in their tweets for questioning about facts or issues that they do not support.

6.5.5 Usefulness of Behavior Types

Lastly, we examine the usefulness of the different behavior types in user classification task. To do this, we perform the same experiments on **One-Week** dataset using the following variants of **CBS** model

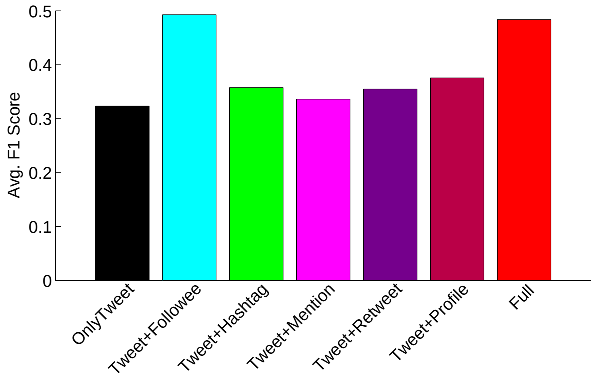


Figure 6.6: **One-Week** dataset: Performance of variants of **CBS** in user classification

- **OnlyTweet**: the variant in which we do not take any behavior (of any type) into account, i.e., only tweets and sentiments are modeled.
- **Tweet+Followee**: the variant in which we only consider tweets, sentiments, and behaviors of *Followee* type. Similarly we have **Tweet+Hashtag**, **Tweet+Mention**, **Tweet+Retweet**, and **Tweet+Profile** variants.
- **Full**: the **CBS** model presented as above where all (5) types of behaviors are taken into account.

Figure 6.6 shows the performance of the different variants of **CBS** in user classification task. The figure suggests that adding behaviors improves the performance, and *Followee* is more useful than other behaviors. We further conducted McNemar's statistical test [102] to compare the variants' performance. The test showed that: (1) the behaviors are helpful in user classification as all the variants with behaviors added have performance that is statistically significantly higher than performance of **OnlyTweet** variant, and (2) among the behaviors, following behavior is the most useful as **Tweet+Followee** and **Full** have statistically significant higher performance than the other variants' performance. The test also showed that the difference between the **Tweet+Followee** and **Full** variants is not statistically significant.

6.6 Chapter Summary

In this chapter, we propose **CBS** model for learning microblogging users' topical interest as well as deriving their community affiliation based on users' content, the sentiments associated with the content, and their behaviors. **CBS** has a novel framework that allows user content and user behavior of different types can be modeled simultaneously. Our experiments on two real Twitter datasets show that the proposed model outperforms baseline methods. The work presented in this chapter was previously published in [82].

Chapter 7

Modeling of Community Behaviors and Content

This chapter presents our work on joint modeling of user and community interests in microblogging using both user content and user behavior. This chapter is organized as follows. In Section 7.1, we first discuss some issues of existing works on modeling the interests that motivated our research. We then state our research objectives and highlight our contributions in Section 7.2. Our proposed model is described in Section 7.3. We describe two experimental datasets and report the results of applying the proposed model on the two datasets in Section 7.4. We report the results of evaluating the proposed model and other topic models in some user profiling tasks in Section 7.5. Finally, we summarize the chapter in Section 7.6.

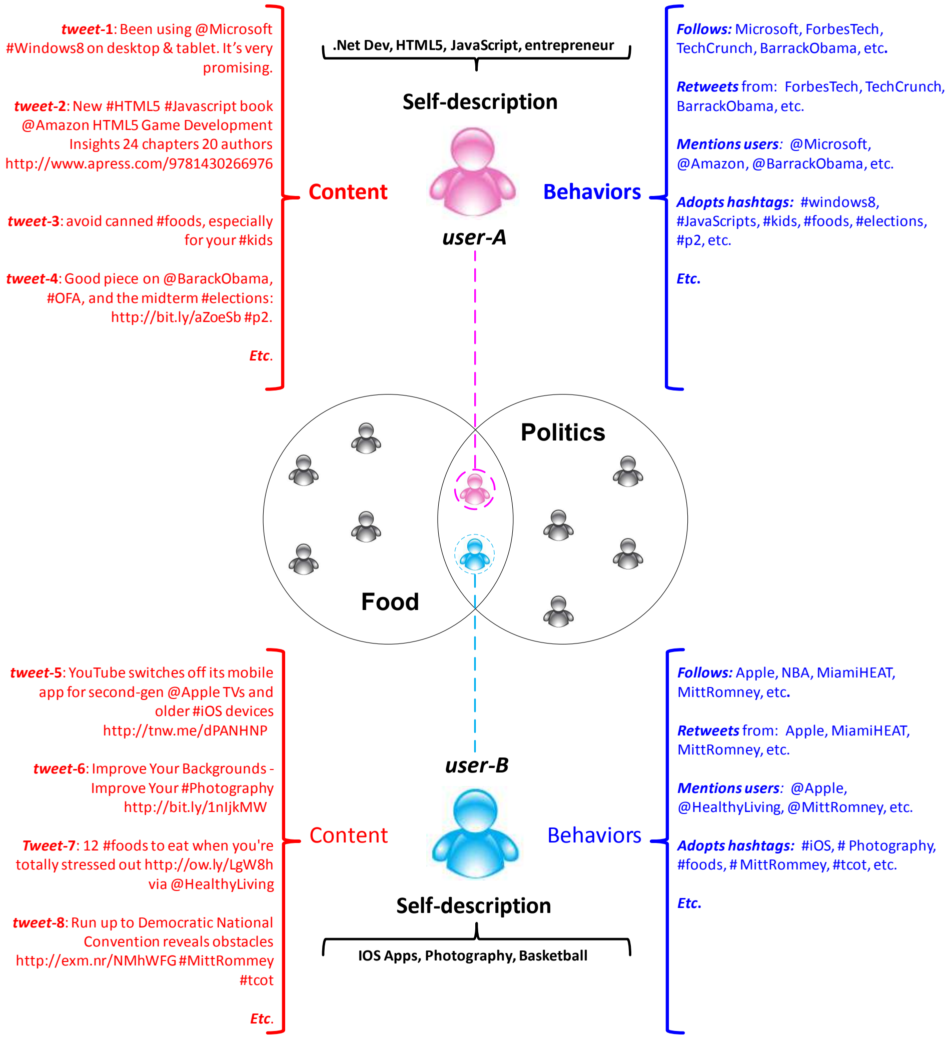
7.1 Motivation

Microblogging users' topical interest and that of their communities have been widely studied. The existing works however suffer from the following two major shortcomings: (i) they do not consider topical communities when modeling users' personal interest, and (ii) they learn users' interest from either their content only or and their behaviors only but not both.

Personal interest and topical communities. Empirical and user studies on microblogging usage have shown that the purpose of tweeting can be broadly attributed to the users' personal topics or background topics [103, 257, 119]. The former cover interests of the users themselves. The latter are the interests shared by users in the same topical communities [71]. Instead of using the term community or social community which usually refers to a social group of densely connected users [174], we use the term *"realm"* to describe a topical user community. Users within a realm may not have many social ties among them, but they share some common background interest. In general, a user can belong to multiple realms. Hence, when modeling microblogging user content and behavior, we have to consider both the users' personal interests and their realms. Previous works however do not consider realms. Some of them do not model background topics at all (e.g.,[92, 179, 242]). Others assume that there is only a single background topic (e.g.,[93, 258, 177, 229]). Without considering realms and background topics, the previous models would not be able to describe the users' personal interest very accurately.

Consider an example in Figure 7.1. There are two realms: *Food* and *Politics*. Both *user-A* and *user-B* belong to the two realms, and therefore they sometime tweet about the realms' topics. For example, *user-A* and *user-B* mention about food in *tweet*-3 and *tweet*-7 respectively, and they also mention about politics in *tweet*-4 and *tweet*-8 respectively. They also adopts the realms' representative behaviors. Being part of the *Food* realm, they use hashtag *#foods*, follow and retweet from *HealthyLiving*[1]. Similarly, they use hashtags *#p2*,*#tcot*,*#elections*,*#MittRomney*, and follow and retweet from *BarackObama*[2], *MittRomney*[3] due to their association with the *Politics* realm. The existing models, in the absence of realms, would treat the two realms' topics as users' personal interests, leading to incorrect personalization decisions.

[1] https://twitter.com/healthyliving
[2] https://twitter.com/barackobama
[3] https://twitter.com/mittromney



tweet-1: Been using @Microsoft #Windows8 on desktop & tablet. It's very promising.

tweet-2: New #HTML5 #Javascript book @Amazon HTML5 Game Development Insights 24 chapters 20 authors http://www.apress.com/9781430266976

tweet-3: avoid canned #foods, especially for your #kids

tweet-4: Good piece on @BarackObama, #OFA, and the midterm #elections: http://bit.ly/aZoeSb #p2.

Etc.

Follows: Microsoft, ForbesTech, TechCrunch, BarrackObama, etc.

Retweets from: ForbesTech, TechCrunch, BarrackObama, etc.

Mentions users: @Microsoft, @Amazon, @BarrackObama, etc.

Adopts hashtags: #windows8, #JavaScripts, #kids, #foods, #elections, #p2, etc.

Etc.

tweet-5: YouTube switches off its mobile app for second-gen @Apple TVs and older #iOS devices http://tnw.me/dPANHNP

tweet-6: Improve Your Backgrounds - Improve Your #Photography http://bit.ly/1nIjkMW

Tweet-7: 12 #foods to eat when you're totally stressed out http://ow.ly/LgW8h via @HealthyLiving

tweet-8: Run up to Democratic National Convention reveals obstacles http://exm.nr/NMhWFG #MittRommey #tcot

Etc.

Follows: Apple, NBA, MiamiHEAT, MittRomney, etc.

Retweets from: Apple, MiamiHEAT, MittRomney, etc.

Mentions users: @Apple, @HealthyLiving, @MittRomney, etc.

Adopts hashtags: #iOS, # Photography, #foods, # MittRommey, #tcot, etc.

Etc.

Figure 7.1: Illustrative example of personal and community interests in microblogging

User content and user behavior. Topical interests determine both content and behaviors of users. For example, in Figure 7.1, *user-A* is interested in Microsoft's .NET framework, HTML5, and entrepreneurship (as stated in her self-description), hence she mentions and retweets from *Microsoft* and *Amazon*; and adopts hashtags like *#windows8*, and *#JavaScripts*. Also, due to topics of her realms, she follows, mentions, and retweets from *BarackObama*, and adopts hashtags like *#kids*, *#food*, *#p2*, and *#elections*. Similarly, *user-B* is interested in IOS applications, and hence mentions and retweets from *Apple*; and adopts the hashtag *#ios*. Also, due to her association with the *Politics* realm, she follows, mentions, and retweets from *MittRomney*, and adopts hashtags like *#food*, *#tcot*, and *#MittRomney*.

To the best of our knowledge, there is no previous work learning users' topical interests using both their content and their behaviors. Most of the existing works either model topics of user content only [92, 258] or user behaviors only [145, 144]. These works neglect the relationship between the two components of microblogging users (i.e., user content and user behaviors) and thus learning the users' interests in a less-than-optimal manner. A user's topical interest may show up in one but not both the components. For example, in Figure 7.1, *user-A* is interested in entrepreneurship motivating him to follow and retweet from *ForbesTech*[4] and *TechCrunch*[5] even though he hardly tweets on entrepreneurship. Similarly, *user-B* is interested in basketball and he follows and retweets from *NBA*[6] and *MiamiHEAT*[7] even though he may not have tweeted about basketball.

Few other works consider both user content and user behaviors together. Sachan *et al.* [194] and Qiu *et al.* [177] model the types of user behavior associated with the content. For example, a message may be associated with behavior types like *tweet* (*post*), *retweet* (*forward*), etc.. These works therefore can only model a subset of user behavior types, and do not model the user behavior instances (e.g., who is retweeted, which hashtag is used, etc.). Aggregating users behaviors by their types is an oversimplification that leads to less accurate models.

7.2 Research Objectives and Contributions

We aim to introduce realms as well as users' topical interest in modeling the content and behavior of microblogging users. We seek to learn realms representing collective topical interests, in addition to users' personal topical interest. We also want to model the user's dependence on the realms to generate

[4] https://twitter.com/ForbesTech
[5] https://twitter.com/TechCrunch
[6] https://twitter.com/NBA
[7] https://twitter.com/MiamiHEAT

both content and behavior.

We first address the modeling of both user content and behavior. A naive approach is to first perform topic modeling on user content and user behavior *separately.* Each user's content can be modeled as a document and we can apply an existing topic model, e.g., LDA [27], to learn the user content topics. For user behavior, we also construct user-behavior documents by considering each user as a document and each behavior she adopted as a word in the document. We then learn the latent topic associating with each of the word using LDA. The drawback of this modeling approach is that we could not establish a natural mapping between the learnt behavior topics and tweet content topics. Moreover, as mentioned above, user topical interest determined purely based on tweets only may not be ideal as a user's topical interest may also show up in his behavior.

Another simple approach for modeling topics of tweets and those of their associated behaviors (e.g., user-mention, hashtag adoption, and retweeting, etc.) is to first perform topic modeling on the tweets, and then assign each user behavior with the topic(s) of its associated tweet(s). For example, for each adopted hashtag h, assign to h the topic(s) of the tweet containing h. This approach however does not work well in the cases where: (1) topic(s) of the tweet cannot be accurately identified due to very short and noisy content; or (2) the topic of the tweet does not fully explain the behavior. For example, Zappavigna *et al.* found that instead of using hashtags to capture topics in tweets, microblogging users have been used hashtags for many other purposes including personalized bookmarking and named entity markup [252].

We therefore propose to jointly model user content and user behavior sharing a common set of latent topics. This approach has several advantages as follows. First, we can learn users' interest using both their content and behaviors. Secondly, it keeps the topics consistent across user content and user behavior, so as to allow user behavior to be semantically interpreted. Thirdly, by integrating user content and behavior through the shared topics, it allows one to make inference of user behavior using the content, and vice versa.

In this chapter, we also want to model realms that capture topical user communities. A simple way to identify the realms is to first perform topic modeling on tweets and user behaviors to find out topical interest of the users, followed by assigning the most common topics among all the users to be the realms' topics. Such an approach however only results in either a single realm including all the popular topics, or a single popular topic for each realm. The approach also does not allow us to quantify, for each user, the degree in which the user depends on realms in tweeting and adopting behaviors. We therefore propose to jointly model user topical interest and realms' topic distributions in the same framework where each user is assigned a parameter to control her bias towards behaving based on her own interest or topical interests of her associated realms.

There are several advantages of modeling realms' topics and user topical interests in a single model. For example, to recommend content to a user, we can directly select content that match user topical interests if the user has little dependence on his realms. For another user who has strong dependence on his realms, we should select content that match the topics of the realms instead. In this way, content recommendation will be more personalized and the messages for the two types of recommendations can also be customized accordingly.

In this chapter, we adopt the "bag-of-words" representation for both user content and user behavior like in Chapter 6. We further develop a new probabilistic graphical model that simultaneously infers latent topics, users' topical interest, and latent realms. Our main contributions in this work consist of the following.

- We propose a probabilistic graphical model, called *Generalized Behavior-Topic* model (abbreviated as **GBT**), for modeling topical interests of users and their realms, as well as for modeling both user content and user behavior using a common set of topics. In **GBT**, the dependency of the users on realms in generating content and adopting behaviors are parameters to be learnt. This is a unique contribution of this work since each user's dependence on realms is not observable in the data.

- We develop a simple sampling method to infer the model's parameters. We further develop an efficient regularization technique to bias the model to learn more semantically clear realms. Our learning method is easy to implement and scales with the number of latent topics and realms, and number of observed content words and behaviors.
- We apply **GBT** model on two Twitter datasets and show that it significantly outperforms state-of-the-art topic models for Twitter content.
- An empirical analysis of topics and realms for the two datasets has been conducted to demonstrate the efficacy of the **GBT** model.
- Lastly, we further demonstrate the application of **GBT** model in some user profiling tasks showing that it also outperforms other topic models in these tasks.

7.3 Generalized Behavior-Topic Model

In this section, we present our proposed *Generalized Behavior-Topic* (**GBT**) model in detail. We first summarize the notations used in this chapter in Table 7.1.

7.3.1 Assumptions

Our model relies on the assumptions that: (i) users generate content and adopt behaviors topically; and (ii) users generate content and adopt behaviors according to either their personal interest or some realms. The first assumption

Table 7.1: Main notations used to describe **GBT** model

$\boldsymbol{\mathcal{U}}/\ \boldsymbol{\mathcal{T}}$	Set of all users/ all tweets
$\boldsymbol{\mathcal{W}}/\ \boldsymbol{\mathcal{B}}$	Bag-of-words/ bag-of-behaviors of all types
$\boldsymbol{\mathcal{V}}_t$	Tweet vocabulary
U/L	Number of users/ Number of behavior types
W	Number of words in tweeting vocabulary
B_l	Number of behaviors of type l
t^i_j/b^{il}_j	j-th tweet/ behavior of type l of user u_i
N_{ij}	#words of tweet t^i_j
w^{ij}_n	n-th word of tweet t^i_j
c^i_j/z^i_j	coin/topic of tweet t^i_j
$c^{i,l}_j/z^{i,l}_j$	coin/topic of behavior $b^{i,l}_j$
K	Number of topics
$\phi_k/\ \lambda_{lk}$	word/ type-l behavior distribution of k-th topic
σ_r	topic distribution of r-th realm
θ_u	topic distribution of user u
$\mu_u/\ \pi_u$	dependence/ realm distribution of user u
$\alpha/\beta/\rho/\tau/\gamma_l$	Dirichlet conjugate priors of $\theta_u/\phi_k/\mu_u/\pi_u/\gamma_l$
$\boldsymbol{\mathcal{C}}/\ \boldsymbol{\mathcal{R}}/\ \boldsymbol{\mathcal{Z}}$	bag-of-coins/ realms/topics of all the tweets and behaviors
$\boldsymbol{\mathcal{C}}_{-t^i_j}/\ \boldsymbol{\mathcal{R}}_{-t^i_j}/\ \boldsymbol{\mathcal{Z}}_{-t^i_j}$	bag-of-coins/ realms/ topics of all behaviors and tweets except t^i_j
$\boldsymbol{\mathcal{C}}_{-b^{i,l}_j}/\ \boldsymbol{\mathcal{R}}_{-b^{i,l}_j}/\ \boldsymbol{\mathcal{Z}}_{-b^{i,l}_j}$	bag-of-coins/ realms/ topics of all behaviors and tweets except $b^{i,l}_j$
$\mathbf{n_c}(c,u,\boldsymbol{\mathcal{C}})$	#times coin c is observed in set of tweets and behaviors of user u for bag-of-coins $\boldsymbol{\mathcal{C}}$
$\mathbf{n_{zu}}(z,u,\boldsymbol{\mathcal{Z}})$	#tweets + #behaviors of user u that have coin 0 and have topic z for bag-of-topics $\boldsymbol{\mathcal{Z}}$
$\mathbf{n_{zr}}(z,r,\boldsymbol{\mathcal{Z}},\boldsymbol{\mathcal{R}})$	#tweets + #behaviors that have coin 1 and have topic z and realm r for bag-of-topics $\boldsymbol{\mathcal{Z}}$, and bag-of-realms $\boldsymbol{\mathcal{R}}$
$\mathbf{n_w}(w,z,\boldsymbol{\mathcal{T}},\boldsymbol{\mathcal{Z}})$	#times word w is observed in topic z for set of tweets $\boldsymbol{\mathcal{T}}$ and bag-of-topics $\boldsymbol{\mathcal{Z}}$
$\mathbf{n^l_b}(b,z,\boldsymbol{\mathcal{B}},\boldsymbol{\mathcal{Z}})$	#times type-l behavior b is observed in topic z for bag-of-behaviors $\boldsymbol{\mathcal{B}}$ and bag-of-topics $\boldsymbol{\mathcal{Z}}$

suggests that, for each user, there is always an underlying topic explaining content of every tweet the user posts as well as every behavior she adopts. The second assumption suggests that, while different users generally have different personal interest, their content and adopted behaviors also share some common topics of the realms. Hence, to model users' content and behaviors accurately, it is important to determine realms as well as their own personal interest.

7.3.2 Generative Process

Based on the above assumptions, we propose **GBT** model with a plate diagram shown in Figure 7.2. **GBT** selects tweet words from a vocabulary $\boldsymbol{\mathcal{V}}_t$; and model behaviors of one of L types where each type-l behavior is drawn from a set of values denoted by $\boldsymbol{\mathcal{V}}_{bl}$. The **GBT** model has K latent topics, where each topic k has (i) a multinomial distribution ϕ_k over the vocabulary $\boldsymbol{\mathcal{V}}_t$, and (ii) a multinomial distribution λ_{lk} over $\boldsymbol{\mathcal{V}}_{bl}$ for each type-l of behaviors. To model realms, **GBT** assumes that there are R realms, where each realm r has a multinomial distribution σ_r over the K topics. Each user u also has a personal topic distribution θ_u over the K topics and a realm distribution π_u over the R realms. Moreover, each user has a dependence distribution μ_u which is a Bernoulli distribution indicating how likely the user behaves based on her own personal interest (μ_u^0) or the realms ($\mu_u^1 = 1 - \mu_u^0$). Lastly, we assume that θ_u, π_u, σ, λ_l, and ϕ have Dirichlet priors α, τ, η, γ_l, and β respectively, while μ_u has Beta prior ρ.

In **GBT** model, we assume the following generative process for all the posted tweets. To generate a tweet t for user u, we first flip a biased coin c_u (whose bias is μ_u) to decide if the tweet will be based on u's personal interest, or one of the realms. If the coin is head up, (i.e., $c_u = 0$), we then choose the topic z_t for the tweet according to u's topic distribution θ_u. Otherwise, (i.e., $c_u = 1$), we first choose a realm r according to u's realm distribution π_u, then we choose z_t according to the chosen realm's topic distribution σ_r. As tweets are short with no more than 140 characters, we assume that each tweet has only one topic. Once the topic z_t is chosen, words in t are then chosen according to the topic's word distribution ϕ_{z_t}. Similarly, we assume the same process for all adopted behaviors, except that, for a behavior b of type l, once the topic z_b is chosen, the behavior is then chosen according to the topic's behavior distribution λ_{lz_b}. The full generative process is as follows.

- For each topic k ($k = 1, \cdots, K$),

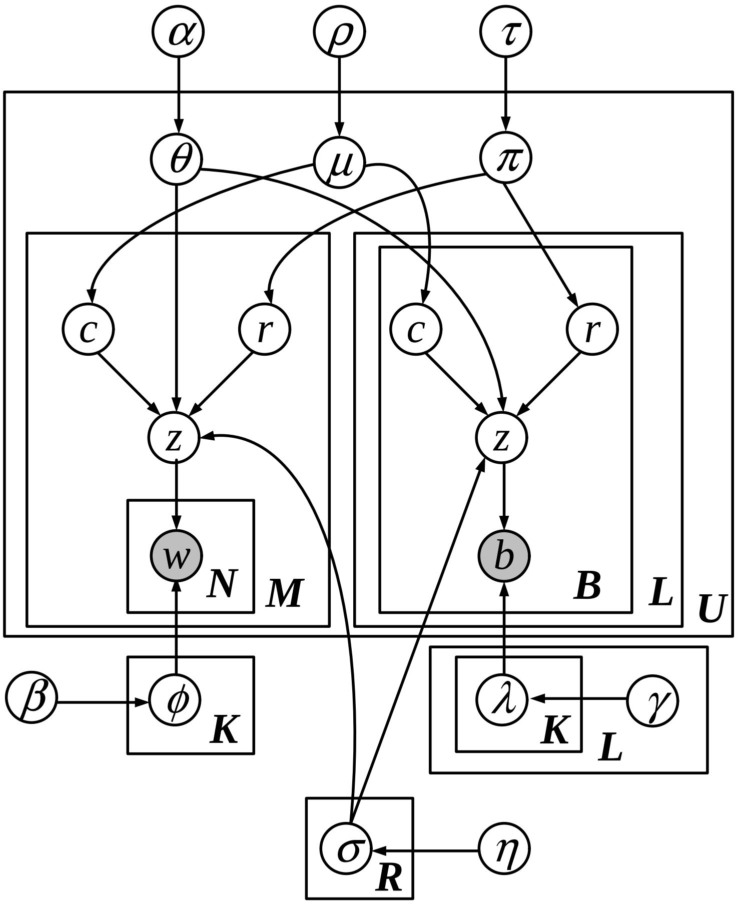


Figure 7.2: Plate notation for **GBT** model

- Sample the topic's word distribution $\phi_k \sim Dirichlet(\beta_k)$
- For each type of behavior l ($l = 1, \cdots, L$), sample the topic's distribution over type-l behaviors $\lambda_{lk} \sim Dirichlet(\gamma_l)$

- For each realm r ($r = 1, \cdots, R$), sample the realm's topic distribution $\sigma_r \sim Dirichlet(\eta_r)$

- For each user u

  1. Sample u's topic distribution $\theta_u \sim Dirichlet(\alpha)$
  2. Sample u's realm distribution $\pi_u \sim Dirichlet(\tau)$
  3. Sample u's dependence distribution $\mu_u \sim Beta(\rho)$

- Generate tweets for the user u: For each tweet t that u posts:

  1. Sample the coin $c_u \sim Bernoulli(\mu_u)$

2. Sample topic for the tweet:
    - if $c_u = 0$, Sample the topic from u's topic distribution: $z_t \sim Multinomial(\theta_u)$
    - If $c_u = 1$, sample the topic from one of the realms:
        * Sample the realm $r_t \sim Multinomial(\pi_u)$
        * Sample the topic $z_t \sim Multinomial(\sigma_{r_t})$
3. Sample the tweet's words: For n-th word of the tweet, sample the word $w_{t,n} \sim Multinomial(\phi_{z_t})$

- Generate behaviors for the user u: For each behavior of type-l that u adopts:

1. Sample the coin $c_u \sim Bernoulli(\mu_u)$
2. Sample topic for the behavior:
    - If $c_u = 0$, sample the topic from u's topic distribution $z_b \sim Multinomial(\theta_u)$
    - If $c_u = 1$, sample the topic from one of the realms
        * Sample the realm $r_b \sim Multinomial(\pi_u)$
        * Sample the topic $z_b \sim Multinomial(\sigma_{r_b})$
3. Sample behavior instance $b \sim Multinomial(\phi_{z_b})$

7.3.3 Model Learning

Consider a set of microblogging users together with their posted tweets and adopted behaviors, we now present the algorithm for performing inference in the **GBT** model. We use U to denote the number of users, and recall that L denotes the number of behavior types in the dataset. We use W to denote the number of words in the tweet vocabulary $\boldsymbol{\mathcal{V}}_t$ (i.e., $W = |\boldsymbol{\mathcal{V}}_t|$), and use B_l to denote the number of behaviors of type l (i.e., $B_l = |\boldsymbol{\mathcal{V}}_{bl}|$). We denote the set of all posted tweets and the bag of all adopted behaviors of all types in the dataset by $\boldsymbol{\mathcal{T}}$ and $\boldsymbol{\mathcal{B}}$ respectively. For each user u_i, we denote her j-th tweet by t_j^i, and denote her j-th behavior of type l by $b_j^{i,l}$. For each posted tweet t_j^i, we denote N_{ij} words in the tweet by $w_1^{ij}, \cdots, w_{N_{ij}}^{ij}$ respectively, and we denote the tweet's topic, coin, and realm (if exists) by z_j^i, c_j^i, and r_j^i respectively. Similarly, for each adopted behavior $b_j^{i,l}$, we denote its topic, coin, and realm (if exists) by $z_j^{i,l}$, $c_j^{i,l}$, and $r_j^{i,l}$ respectively. Lastly, we denote the bag-of-topics, bag-of-coins, and bag-of-realms of all the posted tweets and adopted behaviors in the dataset by $\boldsymbol{\mathcal{Z}}$, $\boldsymbol{\mathcal{C}}$, and $\boldsymbol{\mathcal{R}}$ respectively.

Due to the intractability of LDA-based models [27], we make use of sampling method in learning and estimating the parameters in the **GBT** model. More exactly, we use a collapsed Gibbs sampler ([139]) to iteratively and jointly sample the latent coin and latent realm, and sample latent topic of every posted tweet and adopted behavior.

Sampling for a tweet. For each posted tweet t_j^i, we use $\boldsymbol{\mathcal{C}}_{-t_j^i}$, $\boldsymbol{\mathcal{R}}_{-t_j^i}$, $\boldsymbol{\mathcal{Z}}_{-t_j^i}$ to denote the bag-of-coins, bag-of-realms and bag-of-topics, respectively, of all the adopted behaviors and all other posted tweets in the dataset except the tweet t_i^j. Then the coin c_j^i and the realm r_j^i of t_j^i are jointly sampled according to equations in Figure 7.3, while the topic z_j^i of t_j^i is sampled according to equations in Figure 7.4. Note that when $c_j^i = 0$, we do not have to sample r_j^i, and the current r_j^i (if exists) will be discarded. In these equations, $\mathbf{n_c}(c, u, \boldsymbol{\mathcal{C}})$ records the number of times the coin c is observed in the set of tweets and behaviors of user u for the bag-of-coins $\boldsymbol{\mathcal{C}}$. Similarly, $\mathbf{n_{zu}}(z, u, \boldsymbol{\mathcal{Z}})$ records the number of times the topic z is observed in the set of tweets and the bag of behaviors of user u for the bag of topics $\boldsymbol{\mathcal{Z}}$; $\mathbf{n_{zr}}(z, r, \boldsymbol{\mathcal{Z}}, \boldsymbol{\mathcal{R}})$ records the number of times the topic z is observed in the set of tweets and the bag-of-behaviors that are tweeted/adopted based on the realm r by any user for the bag-of-topics $\boldsymbol{\mathcal{Z}}$ and the bag-of-realms $\boldsymbol{\mathcal{R}}$; $\mathbf{n_{ru}}(r, u, \boldsymbol{\mathcal{R}})$ records the number of times the realm r is observed in the set of tweets and the bag-of-behaviors of user u; and $\mathbf{n_w}(w, z, \boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{Z}})$ records the number of times the word w is observed in the topic z for the set of tweets $\boldsymbol{\mathcal{T}}$ and the bag-of-topics $\boldsymbol{\mathcal{Z}}$.

Figure 7.3: Probabilities used in **jointly sampling coin and realm for tweet** t_j^i without regularization

$$p(c_j^i = 0|\boldsymbol{\mathcal{T}},\boldsymbol{\mathcal{B}},\boldsymbol{\mathcal{C}}_{-t_j^i},\boldsymbol{\mathcal{R}}_{-t_j^i},\boldsymbol{\mathcal{Z}},\alpha,\beta,\tau,\eta,\gamma,\rho) \propto$$

$$\propto \frac{\mathbf{n_c}(0,u_i,\boldsymbol{\mathcal{C}}_{-t_j^i})+\rho_0}{\sum\limits_{c=0}^{1}\Big(\mathbf{n_c}(c,u_i,\boldsymbol{\mathcal{C}}_{-t_j^i})+\rho_c\Big)} \cdot \frac{\mathbf{n_{zu}}(z_j^i,u_i,\boldsymbol{\mathcal{Z}}_{-t_j^i})+\alpha_{z_j^i}}{\sum\limits_{k=1}^{K}\Big(\mathbf{n_{zu}}(k,u_i,\boldsymbol{\mathcal{Z}}_{-t_j^i})+\alpha_k\Big)} \tag{7.1}$$

$$p(c_j^i = 1, r_j^i = r|\boldsymbol{\mathcal{T}},\boldsymbol{\mathcal{B}},\boldsymbol{\mathcal{C}}_{-t_j^i},\boldsymbol{\mathcal{R}}_{-t_j^i},\boldsymbol{\mathcal{Z}},\alpha,\beta,\tau,\eta,\gamma,\rho) \propto \frac{\mathbf{n_c}(1,u_i,\boldsymbol{\mathcal{C}}_{-t_j^i})+\rho_1}{\sum\limits_{c=0}^{1}\Big(\mathbf{n_c}(c,u_i,\boldsymbol{\mathcal{C}}_{-t_j^i})+\rho_c\Big)}$$

$$\cdot \frac{\mathbf{n_{ru}}(r,u_i,\boldsymbol{\mathcal{R}}_{-t_j^i})+\tau_r}{\sum\limits_{r'=1}^{G}\Big(\mathbf{n_{ru}}(r',u_i,\boldsymbol{\mathcal{R}}_{-t_j^i})+\tau_{r'}\Big)} \cdot \frac{\mathbf{n_{zr}}(z_j^i,r,\boldsymbol{\mathcal{Z}}_{-t_j^i},\boldsymbol{\mathcal{R}}_{-t_j^i})+\eta_{rz_j^i}}{\sum\limits_{k=1}^{K}\Big(\mathbf{n_{zr}}(k,r,\boldsymbol{\mathcal{Z}}_{-t_j^i},\boldsymbol{\mathcal{R}}_{-t_j^i})+\eta_{rk}\Big)} \tag{7.2}$$

Figure 7.4: Probabilities used in **sampling topic for tweet** t_j^i without regularization

$$p(z_j^i = z|c_j^i = 0,\boldsymbol{\mathcal{T}},\boldsymbol{\mathcal{B}},\boldsymbol{\mathcal{C}}_{-t_j^i},\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{Z}}_{-t_j^i},\alpha,\beta,\tau,\eta,\gamma,\rho) \propto$$

$$\propto \frac{\mathbf{n_{zu}}(z,u_i,\boldsymbol{\mathcal{Z}}_{-t_j^i})+\alpha_z}{\sum\limits_{k=1}^{K}\Big(\mathbf{n_{zu}}(k,u_i,\boldsymbol{\mathcal{Z}}_{-t_j^i})+\alpha_k\Big)} \cdot \prod_{n=1}^{N_{ij}} \frac{\mathbf{n_w}(w_n^{ij},z,\boldsymbol{\mathcal{Z}}_{-t_j^i})+\beta_{zw_n^{ij}}}{\sum\limits_{v=1}^{W}\Big(\mathbf{n_w}(v,z,\boldsymbol{\mathcal{Z}}_{-t_j^i})+\beta_{zv}\Big)} \tag{7.3}$$

$$p(z_j^i = z|c_j^i = 1,\boldsymbol{\mathcal{T}},\boldsymbol{\mathcal{B}},\boldsymbol{\mathcal{C}}_{-t_j^i},\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{Z}}_{-t_j^i},\alpha,\beta,\tau,\eta,\gamma,\rho) \propto \frac{\mathbf{n_{zr}}(z,r_j^i,\boldsymbol{\mathcal{Z}}_{-t_j^i},\boldsymbol{\mathcal{R}}_{-t_j^i})+\eta_{r_j^i z}}{\sum\limits_{k=1}^{K}\Big(\mathbf{n_{zr}}(k,r_j^i,\boldsymbol{\mathcal{Z}}_{-t_j^i},\boldsymbol{\mathcal{R}}_{-t_j^i})+\eta_{r_j^i k}\Big)} \cdot$$

$$\cdot \prod_{n=1}^{N_{ij}} \frac{\mathbf{n_w}(w_n^{ij},z,\boldsymbol{\mathcal{T}}_{-t_j^i},\boldsymbol{\mathcal{Z}}_{-t_j^i})+\beta_{zw_n^{ij}}}{\sum\limits_{v=1}^{W}\Big(\mathbf{n_w}(v,z,\boldsymbol{\mathcal{T}}_{-t_j^i},\boldsymbol{\mathcal{Z}}_{-t_j^i})+\beta_{zv}\Big)} \tag{7.4}$$

In the right hand side of Equation 7.1: (i) the first term is proportional to the probability that the coin 0 is generated given the priors and (current) values of all other latent variables (i.e., the coins, realms (if exist), and topics of all other tweets and behaviors); and (ii) the second term is proportional to the probability that the (current) topic z_j^i is generated given the priors, (current) values of all other latent variables, and the chosen coin. Similarly, in the right hand side of Equation 7.2: (i) the first term is proportional to the probability that the coin 1 is generated given the priors and (current) values of all other latent variables; (ii) the second term is proportional to the probability that the realm r is generated given the priors, (current) values of all other latent variables, and the chosen coin; and (iii) the third term is proportional to the probability that the (current) topic z_j^i is generated given the priors, (current) values of all other latent variables, and the chosen coin as well as the chosen realm.

In the right hand side of Equation 7.3: (i) the first term is proportional to the probability that the topic z is generated given the priors and (current) values of all other latent variables, and the corresponding coin is 0; and (ii) the second term is proportional to the probability that the tweet content is generated given the priors, (current) values of all other latent variables, and the chosen topic. Similarly, in the right hand side of Equation 7.4: (i) the first term is proportional to the probability that the topic z is generated given the priors and (current) values of all other latent variables, and the and the corresponding coin is 1; (ii) the second term is proportional to the probability that the tweet content is generated given the priors, (current) values of all other latent variables, and the chosen topic.

Sampling for a behavior. Similarly, for each adopted behavior $b_j^{i,l}$, we use $\boldsymbol{\mathcal{C}}_{-b_j^{i,l}}$, $\boldsymbol{\mathcal{R}}_{-b_j^{i,l}}$, $\boldsymbol{\mathcal{Z}}_{-b_j^{i,l}}$ to denote the bag-of-coins, bag-of-realms, and bag-of-topics, respectively, of all the posted tweets and all other adopted behaviors in the dataset except the behavior $b_j^{i,l}$. Then the coin $c_j^{i,l}$ and the realm $r_j^{i,l}$ of $b_j^{i,l}$ are jointly sampled according to equations in Figure 7.5, while the topic $z_j^{i,l}$ of $b_j^{i,l}$ is sampled according to equations in Figure 7.6. Again, note that when $c_j^{i,l} = 0$, we do not have to sample $r_j^{i,l}$, and the current $r_j^{i,l}$ (if exists) will be discarded. In these equations, $\mathbf{n}_{\mathbf{b}}^{\mathbf{l}}(b, z, \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{Z}})$ records the number of times the type-l behavior b is observed in the topic z for the bag-of-behaviors $\boldsymbol{\mathcal{B}}$ and the bag-of-topics $\boldsymbol{\mathcal{Z}}$.

Figure 7.5: Probabilities used in **jointly sampling coin and realm for behavior** $b_j^{l,i}$ without regularization

$$p(c_j^{i,l}=0|\boldsymbol{\mathcal{T}},\boldsymbol{\mathcal{B}},\boldsymbol{\mathcal{C}}_{-b_j^{i,l}},\boldsymbol{\mathcal{R}}_{-b_j^{i,l}},\boldsymbol{\mathcal{Z}},\alpha,\beta,\tau,\eta,\gamma,\rho)\propto$$
$$\propto\frac{\mathbf{n_c}(0,u_i,\boldsymbol{\mathcal{C}}_{-b_j^{i,l}})+\rho_0}{\sum_{c=0}^{1}\Big(\mathbf{n_c}(c,u_i,\boldsymbol{\mathcal{C}}_{-b_j^{i,l}})+\rho_c\Big)}\cdot\frac{\mathbf{n_{zu}}(z_j^{i,l},u_i,\boldsymbol{\mathcal{Z}}_{-b_j^{i,l}})+\alpha_{z_j^{i,l}}}{\sum_{k=1}^{K}\Big(\mathbf{n_{zu}}(k,u_i,\boldsymbol{\mathcal{Z}}_{-b_j^{i,l}})+\alpha_k\Big)}\tag{7.5}$$

$$p(c_j^{i,l}=1,r_j^{i,l}=r|\boldsymbol{\mathcal{T}},\boldsymbol{\mathcal{B}},\boldsymbol{\mathcal{C}}_{-b_j^{i,l}},\boldsymbol{\mathcal{R}}_{-b_j^{i,l}},\boldsymbol{\mathcal{Z}},\alpha,\beta,\tau,\eta,\gamma,\rho)\propto$$
$$\propto\frac{\mathbf{n_c}(1,u_i,\boldsymbol{\mathcal{C}}_{-b_j^{i,l}})+\rho_1}{\sum_{c=0}^{1}\Big(\mathbf{n_c}(c,u_i,\boldsymbol{\mathcal{C}}_{-b_j^{i,l}})+\rho_c\Big)}\cdot\frac{\mathbf{n_{ru}}(r,u_i,\boldsymbol{\mathcal{R}}_{-t_j^i})+\tau_g}{\sum_{r'=1}^{R}\Big(\mathbf{n_{ru}}(r',u_i,\boldsymbol{\mathcal{R}}_{-t_j^i})+\tau_{r'}\Big)}\cdot$$
$$\cdot\frac{\mathbf{n_{zr}}(z_j^{i,l},r,\boldsymbol{\mathcal{Z}}_{-b_j^{i,l}},\boldsymbol{\mathcal{R}}_{-b_j^{i,l}})+\eta_{rz_j^i}}{\sum_{k=1}^{K}\Big(\mathbf{n_{zr}}(k,r,\boldsymbol{\mathcal{Z}}_{-b_j^{i,l}},\boldsymbol{\mathcal{R}}_{-b_j^{i,l}})+\eta_{rk}\Big)}\tag{7.6}$$

Figure 7.6: Probabilities used in **sampling topic for behavior** $b_j^{i,l}$ without regularization

$$p(z_j^{i,l}=z|c_j^{i,l}=0,\boldsymbol{\mathcal{T}},\boldsymbol{\mathcal{B}},\boldsymbol{\mathcal{C}}_{-b_j^{i,l}},\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{Z}}_{-b_j^{i,l}},\boldsymbol{\mathcal{B}},\alpha,\beta,\tau,\eta,\gamma,\rho)\propto$$
$$\propto\frac{\mathbf{n_{zu}}(z,u_i,\boldsymbol{\mathcal{Z}}_{-b_j^{i,l}})+\alpha_z}{\sum_{k=1}^{K}\Big(\mathbf{n_{zu}}(k,u_i,\boldsymbol{\mathcal{Z}}_{-b_j^{i,l}})+\alpha_k\Big)}\cdot\frac{\mathbf{n_b^l}(b_j^{i,l},z,\boldsymbol{\mathcal{B}}_{-b_j^{i,l}},\boldsymbol{\mathcal{Z}}_{-b_j^{i,l}})+\gamma_{lzb_j^{i,l}}}{\sum_{b=1}^{B_l}\Big(\mathbf{n_b^l}(b,z,\boldsymbol{\mathcal{B}},\boldsymbol{\mathcal{Z}}_{-b_j^{i,l}})+\gamma_{lzb}\Big)}\tag{7.7}$$

$$p(z_j^{i,l}=z|c_j^{i,l}=1,\boldsymbol{\mathcal{T}},\boldsymbol{\mathcal{B}},\boldsymbol{\mathcal{C}}_{-b_j^{i,l}},\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{Z}}_{-b_j^{i,l}},\alpha,\beta,\tau,\eta,\gamma,\rho)\propto$$
$$\propto\frac{\mathbf{n_{zr}}(z,r_j^{i,l},\boldsymbol{\mathcal{Z}}_{-b_j^{i,l}},\boldsymbol{\mathcal{R}}_{-b_j^{i,l}})+\eta_{r_j^{i,l}z}}{\sum_{k=1}^{K}\Big(\mathbf{n_{zr}}(k,r_j^{i,l},\boldsymbol{\mathcal{Z}}_{-b_j^{i,l}},\boldsymbol{\mathcal{R}}_{-b_j^{i,l}})+\eta_{r_j^ik}\Big)}\cdot\frac{\mathbf{n_b^l}(b_j^{i,l},z,\boldsymbol{\mathcal{B}}_{-b_j^{i,l}},\boldsymbol{\mathcal{Z}}_{-b_j^{i,l}})+\gamma_{lzb_j^{i,l}}}{\sum_{b=1}^{B_l}\Big(\mathbf{n_b^l}(b,z,\boldsymbol{\mathcal{B}},\boldsymbol{\mathcal{Z}}_{-b_j^{i,l}})+\gamma_{lzb}\Big)}\tag{7.8}$$

The terms in the right hand side of Equations 7.5, 7.6, 7.7, and 7.8 respectively have the same meaning with those of Equations 7.1, 7.2, 7.3, and 7.4.

7.3.4 Sparsity Regularization

As we want to differentiate users' tweets and behavior adoptions based on personal interest from those based on realms while distinguishing one realm from the others, we prefer (a) realms' topic distributions and users' topic distributions to skew on different topics, and (b) different realms' topic distributions to skew on different topics. More exactly, in estimating parameters in the **GBT** model, we need to obtain sparsity in the following distributions.

- Topic specific coin distribution $p^{coin}(\cdot|z)$ where z is a topic: the sparsity in this distribution is to ensure that each topic z is mostly covered by either users' personal interest or realms.
- Topic specific realm distribution $p^{realm}(\cdot|z)$ where z is a topic: the sparsity in this distribution is to ensure that each topic z is mostly covered by one or only a few realms.

To obtain the sparsity mentioned above, we use the *pseudo-observed variable* based regularization technique proposed by Balasubramanyan *et al.* [20] as follows.

7.3.4.1 Topic Specific Coin Distribution Regularization

Since the topic specific coin distributions are determined by both coin and realm joint sampling and topic sampling steps, we regularize both these two steps to bias the distributions to some target sparsity.

In coin and realm joint sampling steps. In each coin & realm sampling step for the tweet t_j^i, we multiply the right hand side of equations in Figure 7.3 with a corresponding regularization term $\boldsymbol{\mathcal{R}}_{\text{topicCoin-Coin\&Realm}}(c|z_j^i)$ which is computed based on empirical entropy of $p(c|z_j^i)$ as in Equation 7.9. Similarly, in each coin & realm sampling step for the behavior $b_j^{i,j}$, we multiply the right hand side of equations in Figure 7.5 with a corresponding regularization term

$\mathcal{R}_{\text{topicCoin-Coin\&Realm}}(c|z_j^{i,l})$ which is computed based on empirical entropy of $p(c|z_j^{i,l})$ as in Equation 7.10.

Figure 7.7: **Topic specific coin distribution regularization terms** used in sampling **coin** and/or **realm** for tweet t_j^i and behavior $b_j^{i,l}$

$$\mathcal{R}_{\text{topicCoin-Coin\&Realm}}\left(c|z_j^i\right) = exp\left(- \frac{\left(H^{coin}_{c_j^i=c}\left(z_j^i\right) - \mu_{\text{topicCoin}}\right)^2}{2\sigma^2_{\text{topicCoin}}}\right) \tag{7.9}$$

$$\mathcal{R}_{\text{topicCoin-Coin\&Realm}}\left(c|z_j^{i,l}\right) = exp\left(- \frac{\left(H^{coin}_{c_j^{i,l}=c}\left(z_j^{i,l}\right) - \mu_{\text{topicCoin}}\right)^2}{2\sigma^2_{\text{topicCoin}}}\right) \tag{7.10}$$

In topic sampling steps. In each topic sampling step for the tweet t_j^i, we multiply the right hand side of equations in Figure 7.4 with a corresponding regularization term $\mathcal{R}_{\text{topicCoin-Topic}}(z|t_j^i)$ which is computed based on empirical entropy of $p(c|z)$ as in Equation 7.11. Similarly, in each topic sampling step for the behavior $b_j^{i,j}$, we multiply the right hand side of equations in Figure 7.6 with a corresponding regularization term $\mathcal{R}_{\text{topicCoin-Topic}}(z|b_j^{i,l})$ which is computed based on empirical entropy of $p(c|z)$ as in equations in Figure 7.12.

Figure 7.8: **Topic specific coin distribution regularization terms** used in sampling **topic** for tweet t_j^i and behavior $b_j^{i,l}$

$$\mathcal{R}_{\text{topicCoin-Topic}}\left(z|t_j^i\right) = exp\left(- \sum_{z'=1}^{K} \left[\frac{\left(H^{coin}_{z_j^i=z}\left(z'\right) - \mu_{\text{topicCoin}}\right)^2}{2\sigma^2_{\text{topicCoin}}}\right]\right) \tag{7.11}$$

$$\mathcal{R}_{\text{topicCoin-Topic}}\left(z|b_j^{i,l}\right) = exp\left(- \sum_{z'=1}^{K} \left[\frac{\left(H^{coin}_{z_j^{i,l}=z}\left(z'\right) - \mu_{\text{topicCoin}}\right)^2}{2\sigma^2_{\text{topicCoin}}}\right]\right) \tag{7.12}$$

In Equation 7.9, $H^{coin}_{c_j^i=c}\left(z_j^i\right)$ is the empirical entropy of $p^{coin}(\cdot|z_j^i)$ when $c_j^i = c$; and in Equation 7.10, $H^{coin}_{c_j^{i,l}=c}\left(z_j^{i,l}\right)$ is the empirical entropy of $p^{coin}(\cdot|z_j^{i,l})$ when $c_j^{i,l} = c$. Similarly, for each topic z', in Equation 7.11, $H^{coin}_{z_j^i=z}\left(z'\right)$ is the empirical entropy of $p^{coin}(\cdot|z')$ when $z_j^i = z$, and in Equation7.12, $H^{coin}_{z_j^{i,l}=z}\left(z'\right)$ is the empirical entropy of $p^{coin}(\cdot|z')$ when $z_j^{i,l} = z$. The two parameters $\mu_{\text{topicCoin}}$ and $\sigma_{\text{topicCoin}}$ are the target mean and target variance of the entropy of $p(c|z)$

respectively. These target mean and target variances are pre-defined parameters. Obviously, these regularization terms (1) increase weight for values of c, r, and z that give lower empirical entropy of $p(c|z)$, and hence increasing the sparsity of these distributions; but (2) decrease weight for values of c, r, and z that give higher empirical entropy of $p(c|z)$, and hence decreasing the sparsity of these distributions.

7.3.4.2 Topic Specific Realm Distribution Regularization

Similarly, since the topic specific realm distributions are determined by both coin & realm joint sampling and topic sampling steps, we regularize both these two steps to bias the distributions to some target sparsity.

In coin & realm joint sampling steps. In each coin & realm sampling step for the tweet t_j^i, we also multiply the right hand side of equations in Figure 7.3 with a corresponding regularization term $\mathcal{R}_{\text{topicRealm-Coin\&Realm}}(c, r|z_j^i)$ which is computed based on empirical entropy of $p(r'|z_j^i)$ as in Equation 7.13. Similarly, in each coin & realm sampling step for the behavior $b_j^{i,j}$, we also multiply the right hand side of equations in Figure 7.5 with a corresponding regularization term $\mathcal{R}_{\text{topicRealm-Coin\&Realm}}(c, r|z_j^{i,l})$ which is computed based on empirical entropy of $p(r'|z_j^{i,l})$ as in Equation 7.14.

Figure 7.9: **Topic specific realm distribution regularization terms** used in sampling **coin** and/or **realm** for tweet t_j^i and behavior $b_j^{i,l}$

$$\mathcal{R}_{\text{topicRealm-Coin\&Realm}}(c, r|z_j^i) = exp\left(-\frac{\left(H^{realm}_{c_j^i=c, r_j^i=r}\left(z_j^i\right) - \mu_{\text{topicRealm}}\right)^2}{2\sigma^2_{\text{topicRealm}}}\right) \quad (7.13)$$

$$\mathcal{R}_{\text{topicRealm-Coin\&Realm}}(c, r|z_j^{i,l}) = exp\left(-\frac{\left(H^{realm}_{c_j^{i,l}=c, r_j^{i,l}=r}\left(z_j^{i,l}\right) - \mu_{\text{topicRealm}}\right)^2}{2\sigma^2_{\text{topicRealm}}}\right) \quad (7.14)$$

In topic sampling steps. In each topic sampling step for the tweet t_j^i, we also multiply the right hand side of equations in Figure 7.4 with a corresponding regularization term $\mathcal{R}_{\text{topicRealm-Topic}}(z|t_j^i)$ which is computed based on empirical entropy of $p(r|z)$ as in Equation 7.15. Similarly, in each topic sampling step for the behavior $b_j^{i,j}$, we multiply the right hand side of equations in Figure 7.6 with a corresponding regularization term $\mathcal{R}_{\text{topicReaml-Topic}}(z|b_j^{i,l})$ which is computed based on empirical entropy of $p(c|z)$ as in equations in Figure 7.16.

Figure 7.10: **Topic specific realm distribution regularization terms** used in sampling **topic** for tweet t_j^i and behavior $b_j^{i,l}$

$$\mathcal{R}_{\text{topicRealm-Topic}}(z|t_j^i) = exp\left(- \sum_{z'=1}^{K} \left[\frac{\left(H_{z_j^i=z}^{realm}(z') - \mu_{\text{topicRealm}} \right)^2}{2\sigma_{\text{topicRealm}}^2} \right] \right) \tag{7.15}$$

$$\mathcal{R}_{\text{topicRealm-Topic}}(z|b_j^{i,j}) = exp\left(- \sum_{z'=1}^{K} \left[\frac{\left(H_{z_j^{i,l}=z}^{realm}(z') - \mu_{\text{topicRealm}} \right)^2}{2\sigma_{\text{topicRealm}}^2} \right] \right) \tag{7.16}$$

In Equation 7.13, $H_{c_j^i=c,r_j^i=r}^{realm}(z_j^i)$ is the empirical entropy of $p^{realm}(\cdot|z_j^i)$ when $c_j^i = c$ & $r_j^i = r$, and in Equation 7.14, $H_{c_j^{i,l}=c,r_j^{i,l}=r}^{realm}(z_j^{i,l})$ is the empirical entropy of $p^{realm}(\cdot|z_j^{i,l})$ when $c_j^{i,l} = c$ & $r_j^{i,l} = r$. Similarly, for each topic z', in Equation 7.15, $H_{z_j^i=z}^{realm}(z')$ is the empirical entropy of $p^{realm}(\cdot|z')$ when $z_j^i = z$, and in Equation 7.16, $H_{z_j^{i,l}=z}^{realm}(z')$ is the empirical entropy of $p^{realm}(\cdot|z')$ when $z_j^{i,l} = z$. The two parameters $\mu_{\text{topicRealm}}$ and $\sigma_{\text{topicRealm}}$ are the target mean and target variance of the entropy of $p(r|z)$ respectively. These target mean and target variances are pre-defined parameters. Obviously, these regularization terms (1) increase weight for values of c, r, and z that give lower empirical entropy of $p(r|z)$, and hence increasing the sparsity of these distributions; but (2) decrease weight for values of c, r, and z that give higher empirical entropy of $p(r|z)$, and hence decreasing the sparsity of these distributions.

7.3.5 Implementation and Complexity

We use two-dimensional tables for keeping the counts $\mathbf{n_c}(c, u, \mathcal{C})$, $\mathbf{n_{zu}}(z, u, \mathcal{Z})$, $\mathbf{n_{zr}}(z, r, \mathcal{Z}, \mathcal{R})$, $\mathbf{n_w}(w, z, \mathcal{T}, \mathcal{Z})$ and $\mathbf{n_b^l}(b, z, \mathcal{B}, \mathcal{Z})$ and call them **counting**

tables. We also use one-dimensional tables for keeping row and column sums of the counting tables and call them **sum tables**; and use one-dimensional tables for keeping the empirical entropies of $p(c|z)$ and $p(r|z)$ and call them **entropy tables**. In each sampling step, only constant time updates on some counting table(s) and sum table(s) are made. For each topic z, the empirical entropies of $p(c|z)$ and $p(r|z)$ are computed based on the row/column z of one of the counting tables. Hence, in each sampling step, the entropy tables can also be updated in constant time as follows. Let E_{current} be the current empirical entropy of $p(r|z)$. E_{current} and is computed from the array $n_1, \cdots, n_R$ which is the row/column z of one of the counting tables, i.e.,

$$E_{\text{current}} = -\sum_{r=1}^{R} \frac{n_r}{\sum_{r=1}^{R} n_r} \log\left(\frac{n_r}{\sum_{r=1}^{R} n_r}\right)$$

Now, assume that n_{r_1} is changed to $n_{r_1} + \Delta$, then the new empirical entropy E_{new} of $p(r|z)$ can be computed from E_{current} as follows.

$$\begin{aligned} E_{\text{new}} &= \frac{1}{\Delta + \sum_{r=1}^{R} n_r}\Bigg[E_{\text{current}}\sum_{r=1}^{R} n_r + \Big(n_{r_1}\log(n_{r_1}) - (n_{r_1}+\Delta)\log(n_{r_1}+\Delta)\Big) + \\ &\qquad + \log(\Delta + \sum_{r=1}^{R} n_r)\Big(\Delta + \sum_{r=1}^{R} n_r\Big) - \Big(\sum_{r=1}^{R} n_r\Big)\log\Big(\sum_{r=1}^{R} n_r\Big)\Bigg] \end{aligned}$$

Given the sum $\sum_{r=1}^{R} n_r$ is kept in a cell of one of the sum tables, the cost of updating the empirical entropy $p(r|z)$ is therefore constant. Similarly, in each sampling step, we can update any entropy table in constant time. Hence, in total, a single iteration of the sampler performs $\mathcal{O}((|\mathcal{W}| + |\mathcal{B}|)(K + R))$ computations where $|\mathcal{W}|$ is the number of observed words and $|\mathcal{B}|$ is the number of observed behaviors in the dataset [79].

In our experiments, we used sampling method with the above sparsity regularization, setting $\mu_{\text{topicCoin}} = \mu_{\text{topicRealm}} = 0$, $\sigma_{\text{topicCoin}} = 0.3$, $\sigma_{\text{topicRealm}} = 0.5$. This corresponds to the case where every topic is assigned to either realms or users' personal interests, and every topic is also assigned to at most one realm. $\sigma_{\text{topicCoin}}$ is set smaller than $\sigma_{\text{topicRealm}}$ so that, for each topic, the topic's coin distribution is more strictly regularized than its realm distribution. We also used conventional symmetric Dirichlet hyperparameters, which are used in previous works (e.g.,[27, 258, 177]). That is, $\alpha = 50/K$, $\beta = 0.01$, $\rho = 2$, $\tau = 1/C$, $\eta = 50/K$, and $\gamma_l = 0.01$ for all $l = 1, \cdots, L$. Given the input dataset, we train the model with 600 iterations of Gibbs sampling. We took 25 samples with a gap of 20 iterations in the last 500 iterations to estimate all the hidden variables.

7.4 Experimental Evaluation

7.4.1 Datasets

Using snowball sampling, we collected the following two datasets for evaluating the **GBT** model.

SE dataset. This dataset is collected from a set of Twitter users who are interested in software engineering. To construct this dataset, we first utilized 100 most influential software developers in Twitter provided in [107] as the seed users. These are highly-followed users who actively tweet about software engineering topics, and they include *Jeff Atwood*[8], *Jason Fried*[9], and *John Resig*[10]. We further expanded the user set by adding all users following at least five seed users so as to get more technology savvy users. Lastly, we took all tweets posted by these users from August 1st to October 31st, 2011 to form the first dataset, called **SE** dataset.

Two-Week dataset. The second dataset is a large corpus of tweets collected just before the 2012 US presidential election. To construct this corpus, we first manually selected a set of 56 *seed users*. These are highly-followed and politically-oriented Twitter users, including major US politicians, e.g., Barack

[8] http://en.wikipedia.org/wiki/Jeff_Atwood

[9] http://www.hanselman.com/blog/AboutMe.aspx

[10] http://en.wikipedia.org/wiki/John_Resig

Table 7.2: Statistics of the experimental datasets used for evaluating GBT model

	SE dataset	**Two-Week** dataset
#user	14,595	24,046
#tweets	3,030,734	3,181,583
#mention adoptions	354,463 (with 2,337 adopters)	653,758 (with 4,628 adopters)
#hashtag adoptions	894,619 (with 3,992 adopters)	1,820,824 (with 9,288 adopters)
#retweet adoptions	909,272 (with 5,324 adopters)	2,396,100 (with 10,576 adopters)

Obama, Mitt Romney, and Newt Gingrich; well known political bloggers, e.g., America Blog, Red State, and Daily Kos; and political sections of US news media, e.g., CNN Politics, and Huffington Post Politics. The set of users was then expanded by adding all users following at least three seed users so as to get more politics savvy users. Lastly, we used all the tweets posted by these users during the two week duration from August 25th to September 7th, 2012 to form the second dataset, known as the **Two-Week** dataset.

We employed the following preprocessing steps to clean both datasets. We first removed stopwords from the tweets. Then, we filtered out tweets with less than 3 non-stopwords. Next, we excluded users with less than 50 (remaining) tweets so as to focus on users with sufficient data. In both the datasets, we consider the following behavior types (1) *mention*, and (2) *hashtag*, and (3) *retweet*. These are messaging behaviors beyond content generation that users may adopt multiple times. Lastly, for each behavior instance, we filtered away those with less than 10 adopting users; and for each user and each type of behaviors, we filtered out all the behaviors if the user adopted less than 50 behaviors of the type. These minimum thresholds are necessary so that, for each behavior and each user, we have enough number of adoption observations for learning both influence of the user's personal interest and that of the realms on behavior adoption.

Table 7.2 shows the statistics of the two datasets after the preprocessing steps. As shown in the table, the two datasets after the filtering are still large. In **SE** dataset, there are about 200 tweets, 150 mention adoptions, 225 hashtag adoptions, and 170 retweet adoptions per user. In **Two-Week** dataset, there are about 120 tweets, 140 mention adoptions, 195 hashtag adoptions, and 225 retweet adoptions per user. This large size allows us to learn the latent factors accurately.

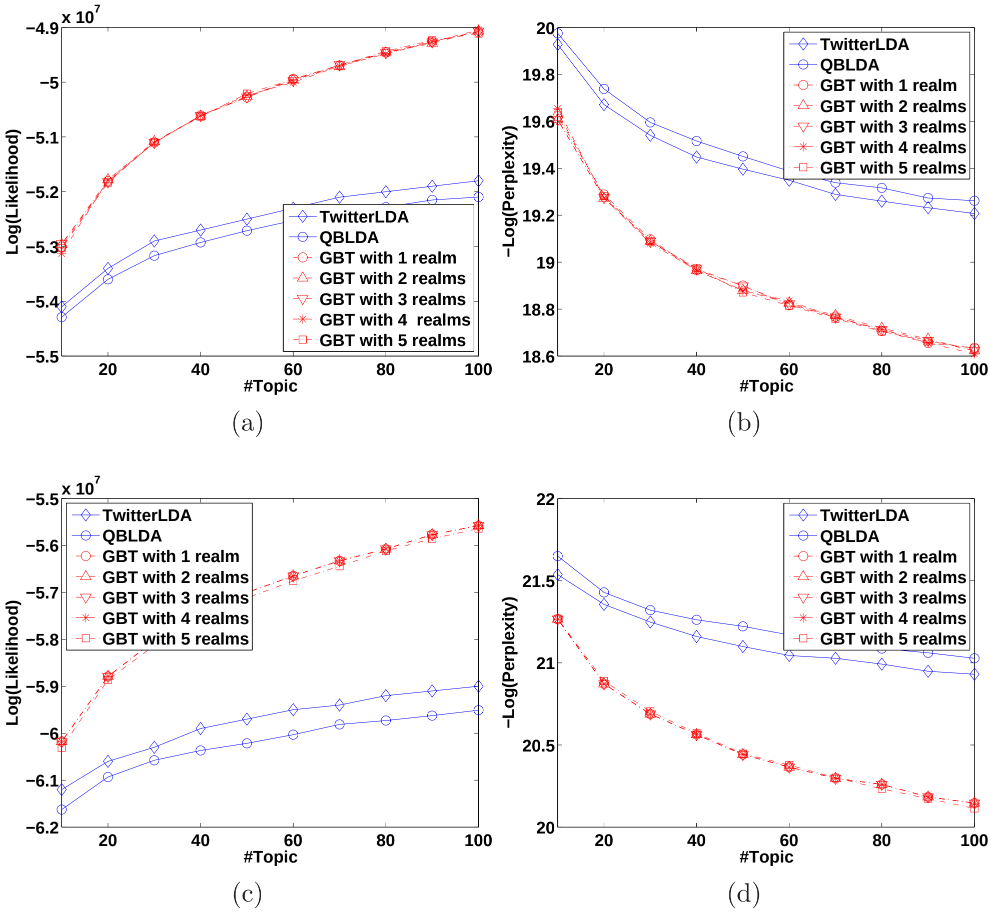
7.4.2 Content Modeling

We first evaluate the ability of **GBT** model in modeling topics of content. To do this, we compare **GBT** with two state-of-the-art topic models for Twitter data: **TwitterLDA** model [258], and **QBLDA** model [177].

We adopt *likelihood* and *perplexity* for evaluating the resultant topics modeling task. For each user, we randomly selected 90% of tweets of the user to form a training set, and the remaining 10% of the tweets as the test set. Then for each model, we compute the likelihood of the training set and perplexity of the test set. The model with a higher likelihood, or lower perplexity is considered better for the task.

Figures 7.11 (a) and (b) show the performance of **TwitterLDA**, **QBLDA** **GBT** models in content modeling on **SE** dataset by varying the number of topics K and the number of realms R. Figures 7.11 (c) and (d) show the similar results on **Two-Week** dataset. As expected, larger number of topics K gives larger likelihood and smaller perplexity, and the amount of improvement diminishes as K increases. The figures show that: (1) **GBT** significantly outperforms both **TwitterLDA** and **QBLDA** models in the content modeling task; and (2) **GBT** is robust against the number of realms as its quantitative performance does not significantly change as we increase the number of the realms from 1 to 5.

We further look into the realms returned by the **GBT** model with different number of realms and found that there is a semantically hierarchical structure among the realms. That is, when the number of realms is increased, the realms are divided into more semantically distinctive realms. For example, Figure 7.12 shows the top topics of the realm(s) found in the **SE** dataset when



(a) (b)

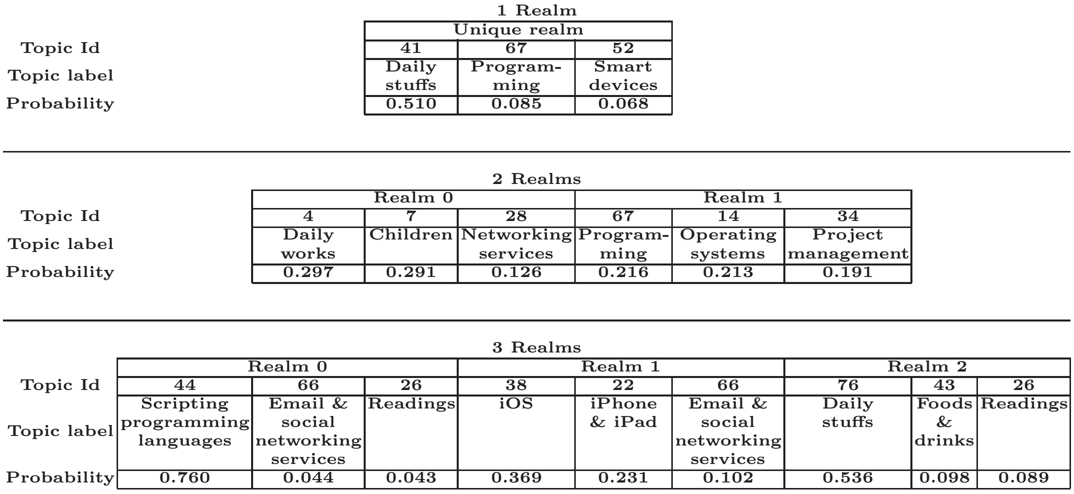
(c) (d)

Figure 7.11: Loglikelihood and Perplexity of **GBT** and TwitterLDA models in: ((a) and (b)) **SE**, and ((c) and (d)) **Two-Week** datasets

the number of realms varied from 1 to 3. Here, the labels of the topics are manually assigned after examining the topics' top words and top tweets. For each topic, the topic's top words are the words having the highest likelihoods given the topic, and the topic's top tweets are the tweets having the lowest perplexities given the topic. The figure clearly shows that the unique realm in the case $R = 1$ is divided into two semantically clearer realms when $R = 2$. These two realms divided into three realms with even clearer semantics when $R = 3$. We also have similar qualitative findings from the **Two-Week** dataset. This suggests that the **GBT** model can recover the more detail realms by increasing the number of realms, even though the quantitative performance does not significantly improve.

Considering both time and space complexities, and it is not practical to

Figure 7.12: Top topics of **realm(s)** found in **SE** dataset when the number of realms varies from 1 to 3



1 Realm

	Unique realm		
Topic Id	41	67	52
Topic label	Daily stuffs	Program-ming	Smart devices
Probability	0.510	0.085	0.068

2 Realms

	Realm 0			Realm 1		
Topic Id	4	7	28	67	14	34
Topic label	Daily works	Children	Networking services	Program-ming	Operating systems	Project management
Probability	0.297	0.291	0.126	0.216	0.213	0.191

3 Realms

	Realm 0			Realm 1			Realm 2		
Topic Id	44	66	26	38	22	66	76	43	26
Topic label	Scripting programming languages	Email & social networking services	Readings	iOS	iPhone & iPad	Email & social networking services	Daily stuffs	Foods & drinks	Readings
Probability	0.760	0.044	0.043	0.369	0.231	0.102	0.536	0.098	0.089

expect a large number of topics falling in realm(s), we set the number of topics to 80 and set the number of the realms to 3 for the experiments presented the following sections.

7.4.3 Background Topics & Realms Analysis

We now examine the background topics found by the **TwitterLDA** and **QBLDA** models, and realms found by the **GBT** model.

Tables 7.3 and 7.4 respectively show the top words of the **background topics** found by **TwitterLDA** model and **QBLDA** model in **SE** dataset, while Table 7.5 shows the top topics for each realm found in the same dataset. Remind that, other than the **background topics** in **TwitterLDA** and **QBLDA** models, the labels of other topics are also manually assigned after examining the topics' top words (shown in Tables 7.13) and top tweets. The label of each realm is also manually assigned after examining the realm's top topics. The tables show that: (i) the **background topics** found by **TwitterLDA** and **QBLDA** models are not semantically clear; while (ii) the realms and their extreme topics found by **GBT** model are both semantically clear and reasonable. In **SE** dataset, other than *Daily Life* realm as reported in [103], it is expected that professional realms *Software Development* and *Apple's product* exist in the dataset as most of its users are working in IT industry. This agrees with the findings by Zhao *et al.* [257] that people also use Twitter for gathering and sharing useful information for their profession.

Table 7.3: Top words of **background topic** found in **SE** dataset by **TwitterLDA** model

life,making,video,blog,change,reading,job,home,thought,line
team,power,game,business,money,friends,talking,starting,month,company

Table 7.4: Top words of **background topic** found in **SE** dataset by **QBLDA** model

video,life,blog,change,job,game,reading,business,power,making
thought,line,home,#fb,giving,friends,team,money,talking,running

Table 7.5: Top topics of **realms** found in **SE** dataset

Realm Id	Realm Label	Top topics		
		Topic Id	Topic Label	Probability
0	Software development	44	Scripting programming languages	0.760
		66	Email & social networking services	0.044
		26	Readings	0.043
1	Apple's products	38	iOS	0.369
		22	iPhone & iPad	0.231
		66	Email & social networking services	0.102
2	Daily life	76	Daily stuffs	0.536
		43	Foods & drinks	0.098
		26	Readings	0.089

Similarly, Tables 7.6 and 7.7 respectively show the top words of the **background topics** found by **TwitterLDA** and **QBLDA** models in **Two-Week** dataset, while Table 7.8 shows the top topics for each realm found in the same dataset. Again, the topics' labels are manually assigned based on examining the topics' top words (shown in Tables 7.19) and top tweets; and the realms' labels are also manually assigned based on examining the realm's top topics. Also, the tables show that: (i) the **background topics** found by **TwitterLDA** and **QBLDA** models are not semantically clear; while (ii) the realms and their extreme topics found by **GBT** model are both semantically clear and reasonable. In **Two-Week** dataset, it is expected that political realms *Responses to DNC & RNC 2012*, *Republicans Opposing*, and *DNC and RNC 2012* exist in the dataset as it was collected during the 2012 US presidential election including the national conventions of both democratic[11] and republican[12] parties.

Table 7.6: Top words of **background topic** found in **Two-Week** dataset by **TwitterLDA** model

life,making,home,america,called,house,change,thought,video,talking
line,american,money,country,job,obama,friends,fact,lost,hell

Table 7.7: Top words of **background topic** found in **Two-Week** dataset by **QBLDA** model

video,making,american,called,obama,america,talking,thought,house,country
president,job,line,giving,home,life,lost,fact,#dnc2012,change

Table 7.8: Top topics of **realms** found in **Two-Week** dataset

Realm Id	Realm Label	Top topics		
		Topic Id	Topic Label	Probability
0	Responses to DNC & RNC 2012	5	Responses to speeches at DNC 2012	0.624
		17	Clint Eastwood's empty chair[13]	0.105
		28	Economics issues	0.072
1	Republicans opposing	8	Criticizing Obama	0.347
		65	Goverment & people's rights	0.138
		3	Criticizing Chris Mathews' comments on Republicans	0.098
2	DNC & RNC 2012	31	Speeches at RNC 2012	0.353
		54	Media reports on DNC & RNC 2012	0.174
		77	Speeches at DNC 2012	0.152

In summary, the empirical content analysis results look reasonable when our proposed **GBT** model is applied on the two datasets. We now turn our focus to behavior modeling results.

7.4.4 User Behavior Analysis

Lastly, we examine the user behaviors associated with the result topics. Tables 7.13 and 7.19 show some of representative topics found in **SE** and **Two-Week** datasets respectively, together with the topics' top behaviors. For each

[11] http://en.wikipedia.org/wiki/2012_Democratic_National_Convention

[12] http://en.wikipedia.org/wiki/2012_Republican_National_Convention

[13] http://en.wikipedia.org/wiki/Clint_Eastwood_at_the_2012_Republican_National_Convention

topic, similar to the topic's top words, the topic's top behaviors are the behaviors having the highest likelihoods given the topic. The tables show that the key behaviors for each of the topics are reasonable. For example, in **SE** dataset, we observe for topic *Scripting programming languages* (topic 44) people use scripting languages related hashtags (*#javascript*, *#ruby*, *#nodejs*, *#php*, etc.), mention and retweet from software project hosting services and scripting language builder & developers (*github*, *@heroku*, *@rubyrogues*, *@steveklabnik*, *garybernhardt*, *tenderlove*, *dhh*, etc.). We also observe for topic *iPhone & iPad* (topic 22) people use iPhone and iPad related hashtags (*#iphone*, *#iphone5*, *#apple*, etc.), mention big IT companies and phone and tablet producers (*branch*, *@twitter*, *@google*, *@amazon*, *@att*, etc.), and retweet from iOS developers and IT blogers(*marcoarment*, *John Gruber*, *dcurtis*).

Similarly, in **Two-Week** dataset, we observe for topic *Reponses to DNC & RNC 2012* (topic 5) people use DNC & RNC 2012 related hashtags (*#dnc2012*, *#rnc2012*, *#literally*, etc.), mention key persons in the two conventions (e.g., *dwstweets*, *stefcutter*, *reince*, etc.), and retweet from political bloggers and commentators (*guypbenson*, *jimgeraghty*, *iowahawkblog*, *jonahnro*, etc.). We also observe for topic *Criticizing Obama* (topic 8)people use negative hashtags related to Obama and DNC 2012 (*#dncin4words*, *#howtopissoffademocrat*, *#overheardatdnc2012*, *#obamatvshows*, etc.), mention and retweet from republican politicians and media (e.g., *@jjauthor*, *@klsouth*, *slone*, *polarcoug*, etc.). A qualitatively similar result holds for the remaining topics as well as topics that are not shown in the two tables.

On the whole, the user behavior analysis results are pretty consistent with that of content analysis. Now that the topics learnt by **GBT** are reasonable, they can be used in the user profiling experiments.

7.5 Utility of User Topics in User Profiling Tasks

In this section, we compare and contrast topics and users' personal topical interests uncovered by **GBT** model with those uncovered by **TwitterLDA** and **LDA** in some user profiling tasks for Twitter. Our aim here is not to propose any new user profiling models. Instead, we want to evaluate the utility of different topic models in the user profiling tasks that differentiate users with different user labels. Here the user labels are the professional and political preferences of the users.

7.5.1 Profiling Tasks

We consider the following tasks.

- **User clustering**. In this task, we use K-mean method with euclidean similarity to cluster a set of users.
- **User classification**. In this task, we use SVM method with linear kernel to classify a set of users into classes corresponding to different user labels.

7.5.2 User Representation

We represent each user by her topic distribution(s) learnt from her content and behaviors using a topic model. More precisely, for each model, each topic is a feature to represent users, and the feature vector of a user is her topic distribution(s) learnt by the model. We examine the following topics models.

- **TwitterLDA**: In this model, each user u is represented by $\theta_u^{TwitterLDA}$ where $\theta_u^{TwitterLDA}$ is the topic distribution of u learnt by **TwitterLDA** model. That means, each user is represented by personal interest learnt from her content only.

- **QBLDA**: In this model, each user u is represented by θ_u^{QBLDA} where θ_u^{QBLDA} is the topic distribution of u learnt by **QBLDA** model. Each user is represented by personal interest learnt from her content and user behavior types associated with the content.

- **TwitterLDA+behaviorLDA**: In this model, we consider both (i) the user's personal interest learnt from her content; and (ii) the user's personal interest that are independently learnt from her behaviors. That means, each user u is represented by a vector feature $\vec{u}$ where $\vec{u}$ is formed by concatenating $\theta_u^{TwitterLDA}$ and $\theta_u^1, \cdots, \theta_u^L$ where θ_u^l is the topic distribution of u learnt by applying LDA [27] on the bags-of-behaviors of type l of all the adopting users if u has type-l behaviors, or a zeros vector otherwise ($l = 1, \cdots, L$). We suppose that adding latent factors learnt from behavior to the **TwitterLDA** model will improve the performance in user profiling tasks.

- **GBT-noBehavior**: For this model, we represent each user u by $\theta_u^{GBT-noBehavior}$ where $\theta_u^{GBT-noBehavior}$ is the topic distribution of u learnt by running **GBT** only on the dataset excluding all user behaviors. With this model, we want to evaluate the effectiveness of user behaviors in profiling a user.

- **GBT-noRegularization**: For this model, we represent each user u by $\theta_u^{GBT-noRegularization}$ where $\theta_u^{GBT-noRegularization}$ is the topic distribution of u learnt by running **GBT** on the full dataset (both user content and user behaviors) but without any sparsity regularization. With this model, we want to evaluate the effectiveness of the proposed sparsity regularization technique in learning clearer user interests.

- **GBT**: For this model, we represent each user u by θ_u^{GBT} where θ_u^{GBT} is topic distribution of u learnt by running **GBT** model on the full dataset and with the regularization technique used. We expect **GBT** to outperform all the previous models. This improvement attributes to: (a) joint modeling of user interest from both user content and user behaviors; and (b) more accurate measuring of users' personal interest after filter out their dependency on realms.

Similarly to the previous experiments, in all the above models, we set the number of topics to 80; and in **GBT-noBehavior**,**GBT-noRegularization** and **GBT** models, we set the number of realms to 3.

7.5.3 Experimental Datasets

To evaluate the performance of the above topic models in user profiling tasks, we need some datasets with ground truth labels for all users. Since we do not have ground truth labels for all users in **SE** and **Two-Week** datasets, we derived following (sub) datasets.

- **Developer** dataset: From the users' *self-descriptions*, we were able to manually label 691 users in **SE** dataset as developers. Among these users, 328 users declare .NET-based programming languages (e.g., C#, Visual Basic, etc.) as their preferential languages, and 363 users declare other languages (e.g., Java, PhP, Python, etc.). We respectively denote the label for the former and latter set of these users by **.NET** and **non-.NET**. Then, for clustering task, we cluster the developers into two clusters. For the classification task, we performed a binary classification.

- **Political affiliation** dataset: Similarly, from users' *self-descriptions*, we were able to manually label 186 users in **Two-Week** dataset as **Democrat** and 1288 users as **Republican**. Again, for clustering task, we cluster these manually labeled users into two clusters; and for the classification task, we also performed a binary classification.

7.5.4 Evaluation Metrics

For convenient, in **Developer** dataset, we call **.NET** user label 1 and call **non-.NET** user label 2. Also, in **Political affiliation** dataset, we call **Democrats** user label 1 and call **Republicans** user label 2.

For user clustering task, we adopt *weighted entropy* as the performance metric. After running K-means method with the number of clusters set to 2, we computed the weighted entropy of the resultant clusters as follows.

$$E = -\sum_{c=1}^{2} \frac{n_c}{N_u} * \left[\frac{n_c^{G1}}{n_c} * log\frac{n_c^{G1}}{n_c} + \frac{n_c^{G2}}{n_c} * log\frac{n_c^{G2}}{n_c}\right] \quad (7.17)$$

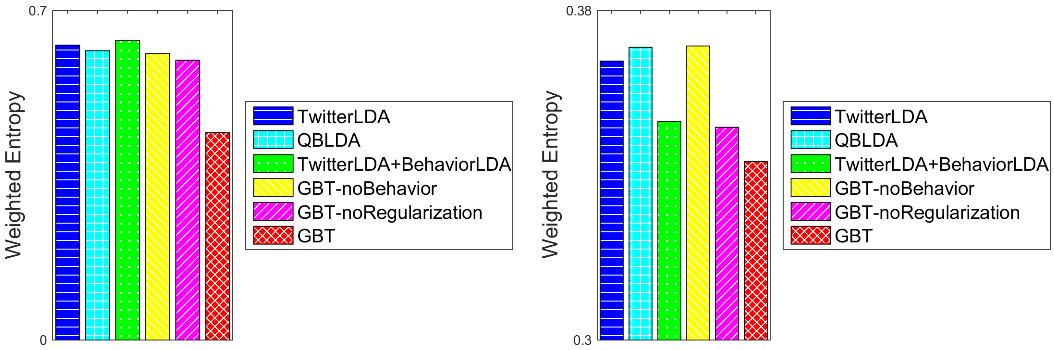
where n_c is the number of users assigned to cluster c, n_c^{G1} and n_c^{G2} is respectively the number of users having user label 1 and user label 2 that are assigned to clustering c; and N_u is total number of users of both the labels. The model with a lower entropy is the winner in the task.

For user classification task, we adopt *average* $F1$ *score* as the performance metric. To do this on a dataset, we first evenly distributed the set of all users in the dataset into 10 folds such that, for each user label, the folds have the same fraction of users having the label. Then, for each model, we use 9 folds to train a SVM classifier using SVMlight toolbox[14], and use the remaining fold to test the learnt classifier. We then compute the average $F1$ score obtained by each model with respect to both the two user labels. The model with a higher score is the winner in the task.

7.5.5 Performance Comparison

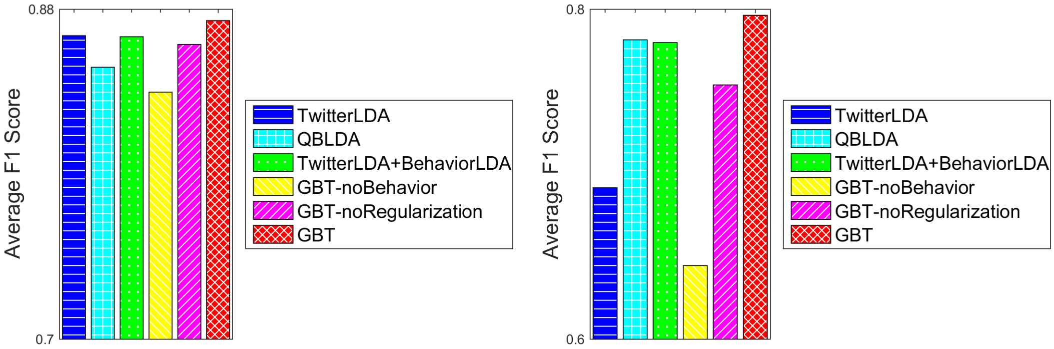
Figure 7.13 shows the weighted entropy of the various models in the user clustering task for the **Developer** and **Political affiliation** datasets. Figure 7.14 shows the average $F1$ scores for the user classification task. The figures show that adding the behavior topic distributions improves the per-

[14] http://svmlight.joachims.org/



(a) (b)

Figure 7.13: Performance of different models in user clustering task in: (a) **Developer**, and (b) **Political affiliation** dataset



(a) (b)

Figure 7.14: Performance of different models in user classification task in: (a) **Developer**, and (b) **Political affiliation** dataset

formance in user profiling. The **TwitterLDA+behaviorLDA** model has lower weighted entropies and higher average $F1$ scores than the **TwitterLDA** model in both the cases. Similarly, the **GBT-noRegularization** and **GBT** models also have lower weighted entropies and higher average $F1$ scores than **GBT-noBehavior** model in both the cases. However, the **QBLDA** model does not always outperform the **TwitterLDA** and **GBT-noBehavior** models. This suggests that, by aggregating user behaviors to their types like in the **QBLDA** model, we may loss useful information for deriving user interest. Lastly, the figures clearly show that **GBT** significantly significantly improves the performance over the **GBT-noRegularization** model, and also significantly outperforms all other models. This implies the effectiveness of the proposed sparsity regularization technique, and the **GBT** model provides a better way for representing users so as to more accurately differentiate users having different preference.

7.5.6 Feature Analysis

Finally, we examine the most representative topic features for each user label learnt by the SVM-based classifiers in the user classification tasks. For each model, we first normalize the topic features' weight returned by the classifiers (in the training phase) by the maximum weight of all the topic features associated with the same model. Hence, for each model, the normalized weight of each topic feature in the model represents the topic's relative importance in the model. As we run 10-fold cross validation, for each model, we compute the average normalized weight of every topic across the 10 folds. The topics with highest and lowest average normalized weights are then the most representative for the two user labels respectively.

Table 7.9: Top representative topics for user label in **Developer** dataset learnt by comparative models

User label	TwitterLDA		QBLDA		TwitterLDA+behaviorLDA		GBT	
	Topic	Topic Label	Topic	Topic Label	Topic	Topic Label	Topic	Topic Label
.NET	66	Microsoft Visual Studio	5	Microsoft Visual Studio	tweet topic 66	Microsoft Visual Studio	69	Microsoft Visual Studio
	7	Windows Tablets & Phones	47	Windows Tablets & Phones	tweet topic 7	Windows Tablets & Phones	35	Windows 8
	40	Lance Armstrong	58	Happenings in London	retweet topic 27	Windows developers	65	Windows Tablets & Phones
non-.NET	75	Data management	79	HTML & Web	tweet topic 75	Data management	44	Scripting programming languages
	47	iOS & iPhone	52	Internet & Media	tweet topic 47	iOS & iPhone	71	Java software development
	64	Entertainment	62	Web Browsers	tweet topic 9	Readings	48	Open-source data management systems

Table 7.9 shows the most representative topics for the two user labels in **Developer** dataset learnt by the comparative models. Again, we manually labeled the topics by examining their top words (as shown in Tables 7.10, 7.11, 7.12, and 7.13) and top tweets. The table clearly shows that

the most representative topics learnt by **GBT** model are more reasonable than the ones learnt by the other models. All the most representative topics learnt by **GBT** model are related to the two programming frameworks (*Microsoft Visual Studio Windows 8*, and *Windows Tablets & Phones* for **.NET** label; and *Scripting programming languages*, *Java software development* and *Open-source data management systems* for **non-.NET** label). On the other hand, the most representative topics for the two user labels learnt by the other models are not always related to the two programming frameworks (e.g., *Entertainment* (**TwitterLDA** model), *Happenings in London* (**QBLDA** model), and *Readings* (**TwitterLDA+behaviorLDA** model)), or semantically discriminative for the frameworks (e.g., *Data management* (**TwitterLDA** and **TwitterLDA+behaviorLDA** models) and *HTML & Web* (**QBLDA** model)).

Table 7.10: Top words of topics discovered by **TwitterLDA** model from **SE** dataset

Topic	Top words
7	windows,microsoft,surface,#windows8,#win8,metro,nokia,xbox,#bldwin,tablet
9	reading,life,internet,book,language,person,english,thought,article,code
40	armstrong,lance,bbc,riot,pussy,police,tour,jones,david,cameron
47	ios,google,iphone,apple,maps,mac,android,ipad,facebook,chrome
64	star,wars,disney,trek,graphics,episode,angry,birds,blog,lucasfilm
66	windows,studio,visual,sharepoint,server,dotnet,sql,#sharepoint,microsoft,azure
75	data,java,node,api,cloud,blog,database,server,code,performance

Table 7.11: Top words of topics discovered by **QBLDA** from model **SE** dataset

Topic	Top words
5	windows,studio,visual,microsoft,azure,sharepoint,#windows8,server,#win8,blog
47	windows,microsoft,surface,nokia,lumia,tablet,xbox,#windows8,tablets,#surface
52	media,science,internet,human,article,reading,data,journalism,change,book
58	bbc,london,police,train,david,british,olympics,boris,olympic,cameron
62	google,maps,ios,apple,chrome,internet,firefox,explorer,microsoft,safari
79	mobile,responsive,content,html5,css,#rwd,device,images,presentation,media

Table 7.12: Top retweeted users of topics discovered by **LDA** model from **SE** dataset

Topic	Top words
27	hmemcpy,hhariri,markrendle,jbogard,adymitruk,gregyoung,troyhunt kellabyte,demisbellot,jeremydmiller

Table 7.13: Top words and top behaviors of topics discovered by **GBT** model from **SE** dataset

Topic	Top words	Top hashtags	Top mentions	Top retweeted
22	iphone,apple,ipad	#iphone,#fail,#iphone5	@branch,@google,@twitter	marcoarment,gruber,dcurtis
	internet,data,wifi	#apple,#win,#appleevent	@kickstarter,@amazon,@att	siracusa,wilshipley,danielpunkass
	home,battery,macbook	#uxaustralia,#iphon,#keynote	@apple,@dropbox,@turf	mrgan,rands,jsnell
26	life,internet,human	#a11y,#a11,#fai	@prismati,@hnycombinator,@prismatic	davewiner,umairh,timoreilly
	problem,media,money	#heweb12,#heweb1,#audio	@leolaporte,@danbenjamin,@doctorow	anildash,pinboard,0xabad1dea
	article,reading,thought	#fail,#accessibility,#facepal	@kevinmarks,@jeffjarvis,@t	cstross,mralancooper,rands
35	windows,microsoft,#windows8	#windows8,#win8,#windows	@microsoft,@surface,@windowsphone	maryjofoley,shanselman,windowsphone
	#win8,hosting,metro	#surface,#bldwi,#wp8	@ch9,@maryjofoley,@windows	thurrott,benthepcguy,gcaughey
	surface,win8,#bldwin	#win,#windowsphone,#wp	@winobs,@windowsazure,@nokia	everythingms,windows,visualstudio
38	ios,mac,iphone	#ios,#android,#ios6	@pocket,@appdotnet,@tweetbot	flyosity,stevestreza,mattgemmell
	apple,google,chrome	#tb,#in,#apple	@marcoarment,@gruber,@tapbots	stroughtonsmith,mantia,panzer
	windows,lion,mountain	#androi,#chrome,#fb	@hotdogsladies,@instagram,@jdalrymple	viticci,sdw,joshhelfferich
43	coffee,beer,eating	#yelp,#sf,#getgluehd	@google,@starbucks,@jason	mike_ftw,paulryangosling,anildash
	dinner,lunch,ice	#opportunity,#sanfrancisco,#chicago	@instagram,@foursquare,@jezebel	pres_bartlet,beep,fakegrimlock
	wine,cream,bacon	#austin,#career,#designer	@gawker,@mike,@kickstarter	joelhousman,pourmecoffee,kissane
44	code,ruby,javascript	#javascript,#ruby,#strangeloo	@github,@heroku,@rubyrogues	steveklabnik,garybernhardt,tenderlove
	git,rails,github	#nodejs,#php,#python	@steveklabnik,@travisci,@madisonruby	dhh,github,roidrage
	python,data,php	#github,#git,#rails	@ashedryden,@simplify,@tenderlove	shit_hn_says,zedshaw,mfeathers
48	data,#bigdata,analytics	#bigdata,#bigdat,#data	@siliconbea,@timoreilly,@harvardbiz	moonpolysoft,shanley,alex_gaynor
	business,google,hadoop	#ibm,#analytics,#hadoop	@whitehouse,@nytimes,@radar	argv0,joedamato,pharkmillups
	information,database,analysis	#ibmiod,#bi,#strataconf	@digiphile,@wired,@slideshare	rickasaurus,jrecursive,cscotta
65	windows,microsoft,nokia	#tech,#technology,#windowsphon	@engadge,@verg,@cne	edbott,verge,tomwarren
	surface,android,tablet	#switchtolumi,#smallbiz,#wincha	@sharethi,@io,@mashabl	drpizza,joshuatopolsky,theromit
	samsung,nexus,lumia	#technews,#microsof,#htc	@youtub,@verge,@rw	ckindel,stroughtonsmith,bdsams
66	email,facebook,google	#facebook,#youtube,#blog	@twitter,@commun,@dropbox	codinghorror,shanselman,rickygervais
	service,spam,emails	#howto,#twitte,#vide	@facebook,@bufferapp,@nealschaffer	levie,codepo8,mattcutts
	password,page,gmail	#google,#lol,#fai	@linkedin,@customerthink,@hootsuite	marscuroisity,morgonfreeman,troyhunt
69	windows,studio,server	#windowsazure,#vs2012,#azure	@pluralsight,@shanselman,@john	shanselman,pluralsight,kellabyte
	visual,sql,dotnet	#sqlserver,#microsoft,#powershell	@shanselma,@codemash,@telerik	jongalloway,haacked,migueldeicaza
	azure,microsoft,blog	#sqlserve,#sql,#mvpbuz	@julielerman,@ch,@oreillymedia	elijahmanor,windowsazure,chrislove
71	sharepoint,java,programming	#sharepoint,#java,#fe	@thefanc,@skillsmatter,@jenkinsci	debasishg,wfaler,dzone
	#sharepoint,code,blog	#javaone,#sp2013,#sharepoin	@dzone,@newsycombinator,@java	fogus,java,psnively
	language,#java,scala	#scala,#sharepoint2013,#javaon	@infoq,@gregyoung,@kevlinhenney	jboner,jamesiry,typesafe
76	home,kids,house	#runkeepe,#wtf,#debat	@klout,@twitter,@runkeeper	neiltyson,sarcasticrover,theonion
	#fb,life,car	#debate,#justsayin,#awesome	@pinteres,@marscuriosity,@kickstarter	robdelaney,wilw,honesttoddler
	dog,room,playing	#awesom,#wt,#winnin	@jack,@theonion,@oatmeal	hotdogsladies,marscuriosity,chrisrockoz

Table 7.14: Top representative topics for user labels in **Political affiliation** dataset learnt by comparative models

User label	TwitterLDA		QBLDA		TwitterLDA+behaviorLDA		GBT	
	Topic	Topic Label	Topic	Topic Label	Topic	Topic Label	Topic	Topic Label
Democrats	35	Romney's taxes and religion	5	Romney's policies on same sex marriage	retweet topic 37 topic 37	Left-leaning political blogers political blogers	7	Romney's tax policy
	2	Democrats on RNC 2012	79	Romney's tax policy	retweet topic 34 topic 34 topic 34	Democrat politicians & pro-democrat organizations	76	Romney's policies on same sex marriage
	30	DNC 2012	36	Voting issues	tweet topic 30	DNC 2012	67	Speeches at DNC 2012
Republicans	10	Republicans on Obama's speech at DNC 2012 at DNC 2012	16	Republicans on Sandra Fluke's speech at DNC 2012	retweet topic 31	Republican politicians & pro-republican organizations	57	Republicans on Sandra Fluke's speech at DNC 2012
	21	Obama's private life	26	Public debt	tweet topic 10	Republicans on Obama's speech at DNC 2012	39	Religion issues
	67	Religion issues speech at DNC 2012	15	Ron Paul	hashtag topic 22	Living status	40	Ron Paul

Similarly, Table 7.14 shows the most representative topics for the two user labels in **Political affiliation** dataset learnt by the comparative models. Also, we manually labeled the topics by examining their top words (as shown in Tables 7.15, 7.16, 7.17, 7.18, and 7.19) and top tweets. Again, the table clearly shows that the most representative topics learnt by **GBT** model are more reasonable than the ones learnt by the two other models. All the representative topics learnt by **GBT** model are related to the two political affiliation labels (*Romneys tax policy* and *Romney's policies on same sex marriage* - where Democrats criticize Romney for his proposed tax policy and his opposing to same sex marriage, and *Speeches at DNC 2012* for the **Democrats** label; *Republicans on Sandra Flukes speech at DNC 2012* - where Republicans angrily react to Sandra Flukes speech at DNC 2012[15], *Religion issues*, and *Ron Paul* for the **Republicans** label). On the other hand, the most representative topics for the two user labels learnt by the two other models are not always representative, e.g., *Romneys taxes and religion* and *Obama's private life* (**TwitterLDA** model) - where people talk about Romney and Obama both positively and negatively; *Voting issues* and *Public debt* (**QBLDA** model) - a topic that was actively talked to by users of both the two parties[16]; and *Living*

[15] http://www.slate.com/blogs/xx_factor/2012/09/06/sandra_fluke_at_the_dnc_angry_reaction_from_the_right_wing_is_good_for_obama_.html

[16] http://www.huffingtonpost.com/2012/08/27/womens-vote-2012-election_n_1832825.html?

status - a controversial topic which was first raised by Republicans followed by many opposing responses even from the Republicans[17].

Table 7.15: Top words of topics discovered by **TwitterLDA** model from **Two-Week** dataset

Topic	Top words
2	#p2,#gop,#tcot,#rnc,#dnc2012,romney,#gop2012,#romney #rnc2012,#obama2012
10	obama,speech,dnc,#dnc2012,stadium,convention,charlotte,#tcot,debt,dems
21	obama,michelle,college,#dnc2012,barack,money,#tcot,president,kids,romney
30	#dnc2012,obama,charlotte,convention,dnc,president,tampa,#dnc delegates,speech
35	romney,mitt,tax,bain,capital,#romney,taxes,money,mormon,#p2
67	god,platform,jerusalem,dnc,party,democrats,#dnc2012,israel,obama,dems

Table 7.16: Top words of topics discovered by **QBLDA** model from **Two-Week** dataset

Topic	Top words
5	gay,marriage,labor,romney,rights,#p2,union,workers,#lgbt
15	paul,ron,romney,gop,#ronpaul,convention,supporters,delegates,rnc
16	fluke,sandra,#dnc2012,bill,jason,clinton,biggs,birth,dnc
26	debt,obama,trillion,#tcot,#dnc2012,#obama,unemployment budget,#romneyryan2012
36	voter,voting,law,federal,ohio,election,texas,gop,voters
79	romney,tax,mitt,bain,taxes,money,rich,cuts,capital

Table 7.17: Top retweeted users of topics discovered by **LDA** model from **Two-Week** dataset

Topic	Top words
34	obama2012,barackobama,truthteam2012,thedemocrats,demconvention michelleobama,donnabrazile,edshow,ofa_nc,jameshaning
37	angryblacklady,otoolefan,gottalaff,shoq,karoli,jeffersonobama,steveweinstein owillis,eclecticbrotha,bobcesca_go

Table 7.18: Top hashtags of topics discovered by **LDA** model from **Two-Week** dataset

Topic	Top words
22	#areyoubetteroff,#failingagenda,#16trillionfail,#areyoubetterof #failingagend,#forward2012,#wirigh,#wiright,#arithmetic,#16trillionfai

[17] http://thecaucus.blogs.nytimes.com/2012/09/04/republicans-ask-are-you-better-off-and-many-reply-yes/

Table 7.19: Top words and top behaviors of topics discovered by **GBT** from **Two-Week** dataset

Topic	Top words	Top hashtags	Top mentions	Top retweeted
3	#tcot,chris,obama msnbc,racist,matthews	#tco,#tcot,#twisters #p2,#twister,#caring	@msnbc,@jasonbiggs,@barackobama @nickelodeontv,@hardball,@sandrafluke	kesgardner,keder,noltenc rbpundit,twitchyteam,iowahawkblog
5	obama,romney,speech #dnc2012,clinton,convention	#dnc201,#dnc2012,#rnc201 #dnc101,#literally,#factcheck	@dwstweets,@chucktodd,@stefcutter @reince,@davidaxelrod,@msnbc	guypbenson,jimgeraghty,iowahawkblog jonahnro,noltenc,melissatweets
7	romney,mitt,tax bain,#p2,capital	#p2,#topprog,#ctl #p2b,#p21,#toppro	@thinkprogres,@dailyko,@dailykos @tp,@thinkprogress,@politicusus	mattison,bluedupage,thenewdeal gottalaff,rcooley123,factsaboutmitt
8	obama,america,president #tcot,romney,barack	#dncin4words,#howtopissoffademocrat #overheardatdnc2012,#obamatvshows	@jjauthor,@klsouth,@slone @katyinindy,@irritatedwoman,@chucknellis	jjauthor,klsouth,nathanhale1775 slone,polarcoug,chucknellis
17	obama,#dnc2012,biden joe,chair,romney	#dnc2012,#rnc2012,#insertchai #insertchair,#overheardatdnc201,#fai	@dloesch,@chrisloesch,@dloesc @clayaiken,@michellemalkin,@katiepavlich	dloesch,soopermexican,gaypatriot cnservativepunk,kurtschlichter,melissatweets
28	obama,debt,jobs #dnc2012,tax,trillion	#dnc2012,#gop2012,#areyoubetteroff #forward2012,#obamaisntworking	@mittromney,@barackobama,@paulryanvp @mittromne,@gop,@thedemocrats	mittromney,paulryanvp,romneyresponse keder,gop,romneycentral
31	#gop2012,romney,#rnc2012 speech,mitt,#rnc	#gop201,#gop2012,#condi #tampa201,#webuiltit,#g0p201	@mittromney,@paulryanvp,@anndromney @gopconvention,@govchristie,@marcorubio	buzzfeedandrew,thefix,zekejmiller chucktodd,daveweigel,ezraklein
39	god,platform,jerusalem dnc,democrats,party	#tcot,#mitt2012,#mitt201 #liblies,#catco,#jcot	@sharethi,@times247,@michellemalki @townhallco,@politic,@hotairblo	dickmorristweet,davidlimbaugh,ingrahamangle michellemalkin,monicacrowley,dennisdmz
40	paul,ron,romney rnc,gop,#ronpaul	#ronpaul,#tlot,#rnc #romney,#gogaryjohnson,#ronpau	@youtub,@govgaryjohnson,@youtube @ronpaul,@dailypau,@cbsradionews	1marchella,govgaryjohnson,tweetamiracle i_am_change_usa,iworkiron,cblacktx
54	convention,#dnc2012,tampa #gop2012,charlotte,rnc	#rnc201,#rnc2012,#tampa #clt,#tampabay,#tampaba	@politico,@nytimes,@ron @washingtonpost,@newtgingrich	antderosa,buzzfeedben,nytjim buzzfeedandrew,gov,daveweigel
57	fluke,clinton,sandra bill,#dnc2012,#tcot	#dnc2012,#waronwomen,#tiot #gop2012,#istandwithann,#breitbartnet	@shareaholi,@newsninja2012 @2016themovie,@sharethis	newsninja2012,tmims50,shaughn_a conservative_vw,thesavvy,becca51178
65	government,america,obama party,god,freedom	#obama,#mittromney,#democrats #gop,#romney,#barackobama	@barackobama,@foxnews,@jjauthor @blackrepublican,@dineshdsouza,@obama	jjauthor,newsninja2012,conservative_vw prfekrdumbrella,2016themovie,pac43
67	clinton,#dnc2012,bill obama,speech,president	#dnc2012,#billclinton,#clinton #dnc,#flotus,#michelleobama	@barackobama,@mittromney @michelleobama,@barackobam	obama2012,barackobama,truthteam2012 thedemocrats,edshow,demconvention
76	rape,gop,gay marriage,platform,akin	#gop,#lgbt,#republican #republicans,#romney,#mitt	@mittromney,@cspanw,@paulryanvp @cspanwj,@anndromney,@reppaulryan	thedailyedge,thenewdeal,barackobama chrisrockoz,rcdewinter,sheshego
77	#dnc2012,#dnc,speech obama,biden,convention	#rn,#rnc,#dn #dnc,#dnc1,#rnc12	@thefix,@ezraklein,@daveweigel @realdonaldtrump,@buzzfeedandrew	mattyglesias,ezraklein,drgrist daveweigel,brianbeutler,pourmecoffee

7.6 Chapter Summary

In this chapter, we propose **GBT** topic model for simultaneously modeling realms and users' topical interest in microblogging data. Our model associates user behaviors with the latent topics and accommodates multiple types of behaviors in a common framework. To learn the model's parameters, we develop an efficient Gibbs sampling method. We further develop a regularization technique incorporating with the sampling method so that the proposed model is biased to learn more semantically clear realms. We also report experiments on two Twitter datasets showing the effectiveness of the proposed model in topic modeling, as well as its improvement over other state-of-the-art topic models in some user profiling tasks. This chapter is a major extension of our work previously published in [86].

Chapter 8

Conclusion and Future Works

8.1 Conclusion

Rich datasets from microblogging sites offer both new research opportunities and challenges. Motivated by many important applications, our research develops models to learn factors that affect content and behavior of microblogging users. Our work consists of two parts: (i) modeling of user behavior in content propagation, and (ii) modeling of individual and community factors in generating content and adopting behavior of multiple types. We summarize the two parts as follows.

The first part includes Chapters 3, 4,and 5. In this part, we define three user and content behavioral factors that drive content propagation behaviors of users, namely, *user virality*, *user susceptibility*, and *content virality.* We develop models for measuring these behavioral factors. The modeling issues here are: (a) inter-relationships among the behavioral factors; (b) missing information about user-content exposure; (c) temporal dynamics of behavioral factors in large microblogging data streams; (d) topic specific behavioral factors; (e) noisy topics in microblogging content; and (f) lack of ground-truth data for evaluation.

In Chapter 3, we address the inter-relationships among users' virality and susceptibility and content virality. We develop a static model that allows us to compute these factors based on their inter-relationships. This model measures the factors using an iterative computation method. To overcome the absence of ground truth, we evaluate the proposed model using synthetically generated datasets. We also evaluate the model results in a hashtag retweet order prediction task using a real dataset.

Next, in Chapter 4, we further extend the model proposed in Chapter 3 to also address the temporal dynamics of the user virality, user susceptibility, and content virality in large data streams, and the missing user-item exposure observations. Moreover, we consider the problem in a more general setting in which users may have multiple adoptions/ propagations on the same content item. To do this, we first develop a static model that utilizes these behavioral factors' inter-dependencies and the item adoption/ propagation counts of users, but does not require knowledge about user-item exposure. We then propose an efficient method for assigning temporal weight to data observations so that less weights are given to older observations. Lastly, to deal with the high computational cost of the temporal model, we incorporate users' propagation rank with the factors' inter-dependencies to develop an incremental model for working with large data streams. We evaluate the proposed models by examining their performance in a future propagation count prediction task. We further evaluate the efficacy of the incremental model by examining its computational cost, both theoretically and empirically.

In Chapter 5, we address the issues in modeling the virality and susceptibility factors specific to topics. We first propose a heuristic method for inferring user-content item exposure. Then, based on the state-of-the-art topic model designed specially for microblogging content, we propose a factorization framework for deriving virality of content topics as well as topic-specific virality of users propagating the content, and topic-specific susceptibility of the users who the content is propagated to. We then develop two factorization models that implement the framework. We conduct a series of experiments to evaluate the framework and its associated models using both real and synthetic datasets. We further examine the performance of the proposed models in a propagation prediction task.

The second part includes Chapters 6, and 7. In this part, we aim to learn users' personal interest and communities from both their behaviors and their content. To do this, we have to address the following issues: (a) multiple types of user behaviors users, and (b) distinguishing users' personal interest from that of their communities.

In Chapter 6, we address the issue in simultaneous modeling of user content and user behavior of different types. We propose to represent both users' content and their behaviors using as "bag-of-words". Coupling with an existing sentiment analysis tool for microblogging content, we then develop a topic model for deriving users' community from their behaviors, content, and the sentiment expressed in their content. For simplicity, we consider the case when user communities are non-overlapping. This allows us to leverage partially labeled users for supervising the model's learning process. Our proposed model therefore can be used as both unsupervised and semi-supervised learning models. We evaluate the model's performance in learning topics in users' content, comparing with the state-of-the-art topic model for microblogging content. We also examine the model's ability in user profiling (i.e., learning of users' community labels) as both a unsupervised and a semi-supervised learner.

Lastly, in Chapter 7, we extend the model proposed in Chapter 6 to also address the issues in simultaneous modeling of users' personal interest and communities' interests. We also consider the general case when user communities are overlapping. We develop a new topic model to learn users' personal interest and that of there communities, as well as users' biased toward the communities when generating content and adopting behavior. In this new model, user content and user behavior are jointly modeled using a set of common latent topics, hence allows us to learn the interests from both the content and behavior. We further develop an effective regularization technique for biasing the model to learn more semantically clear topics and communities. With this regularization, the proposed model significantly outperforms state-of-the-art topic models in both learning topics of user content and some user profiling tasks.

To summarize, our main contribution in this dissertation is in joint modeling of factors concerning microblogging users' behaviors and content. While there is a number of research on these factors, the prior works measure them independently despite they have some inter-relationships. We seek to overcome this shortcoming by developing new methods for measuring the factors that consider their inter-relationships, thus obtain more accurate modeling results.

8.2 Future Works

To conclude this dissertation, we outline below several potential directions for future research that can further improve the current work.

First, the iterative computation method we used in Chapter 3 does not come with a theoretical analysis. It remains to prove that the method always converges to a unique scores for the virality and susceptibility factors regardless of the initialization. Also, it would be worth to investigate the temporal dynamics of the factors specific to topics. This allows us to combine the advantages of both the temporal models (developed in Chapter 4) and topic-specific models (developed in Chapter 5) to design even more effective models.

In modeling content propagation behavior of users, we have been assuming that users' links are casual and identical in strength. Hence, a natural extension is to relax this assumption by incorporating heterogeneous pair-wise social influence among users. It would also important to calibrate more fine-grained factors affecting the propagation. These factors include psychological factors of users and linguistic and sentiment features of content.

Next, we would like to consider the scalability of our proposed models for modeling community behavior. We have been using Gibbs sampling method in learning our models' parameters, which may be computationally expensive when the data is sparse. Possible solutions for scaling up the models are approximated and distributed implementations sampling procedures [159, 130, 249], and stale synchronous parallel implementation of variational inference procedures [81].

A user may adopt a behavior because she is socially or topically motivated [174]. Distinguishing between these two types of motivation is important to many applications but still a challenging problem. We therefore would like to extend the proposed models to also incorporate social factors in modeling user behavior. Examples of these factors are social communities and ego networks of users.

Finally, for a long term goal, modeling user behavior in their socio-phyical contexts is a promising research direction we wish to pursue. We envisage that this research can be greatly extended considering multiple data sources beyond user content and user behavior. Examples of the these sources are geo-information associated with user content, user mobility traces provided by their handheld devices, and other information channels like mass media or blogs.

Bibliography

[1] Palakorn Achananuparp, Ee-Peng Lim, Jing Jiang, and Tuan-Anh Hoang. Who is retweeting the tweeters? modeling, originating, and promoting behaviors in the twitter network. *TMIS*, 2012.

[2] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.

[3] Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9, 2008.

[4] Leman Akoglu, Hanghang Tong, Brendan Meeder, and Christos Faloutsos. Pics: Parameter-free identification of cohesive subgroups in large attributed graphs. In *SDM*, 2012.

[5] Aris Anagnostopoulos, Ravi Kumar, and Mohammad Mahdian. Influence and correlation in social networks. In *KDD '08*, 2008.

[6] Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using pagerank vectors. In *FOCS*, 2006.

[7] Reid Andersen and Kevin J Lang. Communities from seed sets. In *WWW*, 2006.

[8] R. M. Anderson and R. M. May. *Infectious Diseases of Humans: Dynamics and Control*. Oxford University Press, 1992.

[9] Demetres Antoniades and Constantine Dovrolis. Co-evolutionary dynamics in social networks: A case study of twitter. *CoRR*, abs/1309.6001, 2013.

[10] Sinan Aral and Dylan Walker. Identifying influential and susceptible members of social networks. *Science*, 337(6092):337–341, 2012.

[11] Sinan Aral and Dylan Walker. Identifying influential and susceptible members of social networks. *Science*, 2012.

[12] Yoav Artzi, Patrick Pantel, and Michael Gamon. Predicting responses to microblog posts. In *NAACL HLT*, 2012.

[13] Konstantin Avrachenkov, Nelly Litvak, Danil Nemirovsky, and Natalia Osipova. Monte carlo methods in pagerank computation: When one iteration is sufficient. *SIAM J. Num. Ana.*, 2007.

[14] Bahman Bahmani, Abdur Chowdhury, and Ashish Goel. Fast incremental and personalized pagerank. *VLDB Endowment*, 2010.

[15] N T J Bailey. *The mathematical theory of infectious diseases and its applications. 2nd edition.* Griffin, 1975.

[16] Jin Yeong Bak, Suin Kim, and Alice Oh. Self-disclosure and relationship strength in twitter conversations. In *ACL*, 2012.

[17] JinYeong Bak, Chin-Yew Lin, and Alice Oh. Self-disclosure topic model for classifying and analyzing twitter conversations. In *EMNLP*, 2014.

[18] Eytan Bakshy, Itamar Rosenn, Cameron Marlow, and Lada Adamic. The role of social networks in information diffusion. In *WWW*, 2012.

[19] Ramnath Balasubramanyan and William W. Cohen. Block-lda: Jointly modeling entity-annotated text and entity-entity links. In *SDM*, 2011.

[20] Ramnath Balasubramanyan and William W. Cohen. Regularization of latent variable models to obtain sparsity. In *SDM*, 2013.

[21] Ramnath Balasubramanyan and Aleksander Kolcz. w00t! feeling great today! chatter in twitter: Identification and prevalence. In *ASONAM*, 2013.

[22] Roja Bandari, Sitaram Asur, and Bernardo A Huberman. The pulse of news in social media: Forecasting popularity. In *ICWSM*, 2012.

[23] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. Who to follow and why: Link prediction with explanations. In *KDD*, 2014.

[24] Frank M. Bass. A new product growth for model consumer durables. *Management Science*, 15(5):pp. 215–227, 1969.

[25] Jonah Berger and Katherine L Milkman. What makes online content viral? *Journal of Marketing Research*, 2012.

[26] Jingwen Bian, Yang Yang, and Tat-Seng Chua. Predicting trending messages and diffusion participants in microblogging network. In *SIGIR*, 2014.

[27] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *JMLR*, 2003.

[28] Francesco Bonchi, Carlos Castillo, and Dino Ienco. Meme ranking to maximize posts virality in microblogging platforms. *J. Intell. Inf. Syst.*, 2013.

[29] Antoine Boutet, Hyoungshick Kim, and Eiko Yoneki. What's in your tweets? i know who you supported in the uk 2010 general election. In *ICWSM*, 2012.

[30] Tom Broxton, Yannet Interian, Jon Vaver, and Mirjam Wattenhofer. Catching a viral video. *J. Intel. Info. Sys.*, 2013.

[31] Ethem F. Can, Hüseyin Oktay, and R. Manmatha. Predicting retweet count using visual cues. In *CIKM*, 2013.

[32] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In *WWW*, 2011.

[33] Meeyoung Cha, Hamed Haddadi, Fabrício Benevenuto, and P. Krishna Gummadi. Measuring user influence in twitter: The million follower fallacy. In *ICWSM*, 2010.

[34] Charalampos Chelmis and Viktor K Prasanna. Predicting communication intention in social networks. In *SocialCom*, 2012.

[35] Jilin Chen, Rowan Nairn, and Ed Chi. Speak little and well: Recommending conversations in online social streams. In *CHI*, 2011.

[36] Kailong Chen, Tianqi Chen, Guoqing Zheng, Ou Jin, Enpeng Yao, and Yong Yu. Collaborative personalized tweet recommendation. In *SIGIR*, 2012.

[37] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*, 2010.

[38] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *SIGKDD*. ACM, 2009.

[39] Justin Cheng, Lada Adamic, P Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. Can cascades be predicted? In *WWW*, 2014.

[40] Justin Cheng, Daniel M Romero, Brendan Meeder, and Jon Kleinberg. Predicting reciprocity in social networks. In *SocialCom*. IEEE, 2011.

[41] Xueqi Cheng, Xiaohui Yan, Yanyan Lan, and Jiafeng Guo. Btm: Topic modeling over short texts. *TKDE*, 26(12), Dec 2014.

[42] Fan Chung and Linyuan Lu. The average distances in random graphs with given expected degrees. *Internet Mathematics*, 1:15879–15882, 2002.

[43] Robert B Cialdini and Noah J Goldstein. Social influence: Compliance and conformity. *Annu. Rev. Psychol.*, 2004.

[44] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[45] Raviv Cohen and Derek Ruths. Classifying political orientation on twitter: It's not easy! In *ICWSM*, 2013.

[46] Giovanni Comarela, Mark Crovella, Virgilio Almeida, and Fabricio Benevenuto. Understanding factors that affect response rates in twitter. In *HT*, 2012.

[47] Michael Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Alessandro Flammini, and Filippo Menczer. Political polarization on twitter. In *5th ICWSM*, 2011.

[48] Peng Cui, Fei Wang, Shaowei Liu, Mingdong Ou, Shiqiang Yang, and Lifeng Sun. Who should share what?: item-level social influence prediction for users and posts ranking. In *SIGIR '11*, 2011.

[49] Evandro Cunha, Gabriel Magno, Giovanni Comarela, Virgilio Almeida, Marcos André Gonçalves, and Fabrício Benevenuto. Analyzing the dynamic evolution of hashtags on twitter: a language-based approach. In *LSM workshop, ACL*, 2011.

[50] Evandro Cunha, Gabriel Magno, Marcos André Gonçalves, César Cambraia, and Virgilio Almeida. He votes or she votes? female and male discursive strategies in twitter political hashtags. *PloS one*, 9(1), 2014.

[51] Onkar Dabeer, Prachi Mehendale, Aditya Karnik, and Atul Saroop. Timing tweets to increase effectiveness of information campaigns. In *ICWSM*, 2011.

[52] Kushal Shailesh Dave, Rushi Bhatt, and Vasudeva Varma. Modelling action cascades in social networks. In *ICWSM*, 2011.

[53] Gianmarco De Francisci Morales, Aristides Gionis, and Claudio Lucchese. From chatter to headlines: Harnessing the real-time web for personalized news recommendation. In *WSDM*, 2012.

[54] Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *JAsIs*, 1990.

[55] Qiming Diao and Jing Jiang. A unified model for topics, events and users on Twitter. In *EMNLP*, 2013.

[56] Qiming Diao, Jing Jiang, Feida Zhu, and Ee-Peng Lim. Finding bursty topics from microblogs. In *ACL*, 2012.

[57] Ying Ding. Community detection: Topological vs. topical. *Journal of Informetrics*, 5(4):498–514, 2011.

[58] Zhuoye Ding, Xipeng Qiu, Qi Zhang, and Xuanjing Huang. Learning topical translation model for microblog hashtag suggestion. In *IJCAI*, 2013.

[59] Wei Dong, Minghui Qiu, and Feida Zhu. Who am i on twitter?: A cross-country comparison. In *WWW*, 2014.

[60] Albert Feller, Matthias Kuhnert, Timm Oliver Sprenger, and Isabell M. Welpe. Divided they tweet: The network structure of political microbloggers and discussion topics. In *ICWSM*, 2011.

[61] Wei Feng and Jianyong Wang. Retweet or not?: personalized tweet re-ranking. In *WSDM*, 2013.

[62] Wei Feng and Jianyong Wang. We can learn your# hashtags: Connecting tweets to explicit topics. In *ICDE*, 2014.

[63] Dániel Fogaras and Balázs Rácz. Towards scaling fully personalized pagerank. In *Algorithms and Models for the Web-Graph*. 2004.

[64] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[65] Santo Fortunato and Marc Barthélemy. Resolution limit in community detection. *PNAS*, 2007.

[66] Shuai Gao, Jun Ma, and Zhumin Chen. Modeling and predicting retweeting dynamics on microblogging platforms. In *WSDM*, 2015.

[67] Ruth García-Gavilanes, Yelena Mejova, and Daniele Quercia. Twitter ain't without frontiers: Economic, social, and cultural boundaries in international communication. In *CSCW*, 2014.

[68] Sharad Goel, Ashton Anderson, Jake Hofman, and Duncan Watts. The structural virality of online diffusion. *Preprint*, 22:26, 2013.

[69] Jennifer Golbeck, Justin M. Grimes, and Anthony Rogers. Twitter use by the u.s. congress. *J. Am. Soc. Inf. Sci. Technol.*, 2010.

[70] Scott A. Golder and Sarita Yardi. Structural predictors of tie formation in twitter: Transitivity and mutuality. In *SOCIALCOM*, 2010.

[71] Przemyslaw A. Grabowicz, Luca Maria Aiello, Victor M. Eguiluz, and Alejandro Jaimes. Distinguishing topical and social groups based on common identity and bond theory. In *WSDM*, 2013.

[72] Laura A. Granka, Thorsten Joachims, and Geri Gay. Eye-tracking analysis of user behavior in www search. In *SIGIR*, 2004.

[73] Mark Granovetter. Threshold models of collective behavior. *American journal of sociology*, 1978.

[74] Daniel Gruhl, Ramanathan Guha, David Liben-Nowell, and Andrew Tomkins. Information diffusion through blogspace. In *WWW*, 2004.

[75] Marco Guerini, Alberto Pepe, and Bruno Lepri. Do linguistic style and readability of scientific abstracts affect their virality? In *ICWSM*, 2012.

[76] Ádám Gyenge, Janne Sinkkonen, and András A. Benczúr. An efficient block model for clustering sparse graphs. In *MLG Workshop, ICML*, 2010.

[77] John Hannon, Mike Bennett, and Barry Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *RecSys*, 2010.

[78] Lars Kai Hansen, Adam Arvidsson, Finn Aarup Nielsen, Elanor Colleoni, and Michael Etter. Good friends, bad news - affect and virality in twitter. In *Future Information Technology*. 2011.

[79] Gregor Heinrich. Parameter estimation for text analysis. Technical report.

[80] John Allen Hendricks and Robert E. Denton Jr. *Communicator-In-Chief: How Barack Obama Used New Media Technology to Win the White House*. Lexington Books, 2010.

[81] Qirong Ho, James Cipar, Henggang Cui, Seunghak Lee, Jin Kyu Kim, Phillip B Gibbons, Garth A Gibson, Greg Ganger, and Eric Xing. More effective distributed ml via a stale synchronous parallel parameter server. In *NIPS*, 2013.

[82] Tuan-Anh Hoang, William W Cohen, and Ee-Peng Lim. On modeling community behaviors and sentiments in microblogging. *SDM*, 2014.

[83] Tuan-Anh Hoang, William W Cohen, Ee-Peng Lim, Doug Pierce, and David P Redlawsk. Politics, sharing and emotion in microblogs. In *ASONAM*, 2013.

[84] Tuan-Anh Hoang and Ee-Peng Lim. Virality and susceptibility in information diffusions. In *ICWSM*, 2012.

[85] Tuan-Anh Hoang and Ee-Peng Lim. Retweeting: An act of viral users, susceptible users, or viral topics? In *SDM*, 2013.

[86] Tuan-Anh Hoang and Ee-Peng Lim. On joint modeling of topical communities and personal interest in microblogs. In *SocInfo*. 2014.

[87] Tuan-Anh Hoang, Ee-Peng Lim, Palakorn Achananuparp, Jing Jiang, and Feida Zhu. On modeling virality of twitter content. In *ICADL*, 2011.

[88] Thomas Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, 1999.

[89] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[90] Courtenay Honeycutt and Susan C. Herring. Beyond microblogging: Conversation and collaboration via twitter. In *HICSS*, 2009.

[91] Liangjie Hong, Ovidiu Dan, and Brian D Davison. Predicting popular messages in twitter. In *WWW*, 2011.

[92] Liangjie Hong and Brian D. Davison. Empirical study of topic modeling in twitter. In *SOMA '10*, 2010.

[93] Liangjie Hong, Byron Dom, Siva Gurumurthy, and Kostas Tsioutsiouliklis. A time-dependent topic model for multiple text streams. In *KDD*, 2011.

[94] Liangjie Hong, Aziz S Doumith, and Brian D Davison. Co-factorization machines: modeling user interests and predicting individual decisions in twitter. In *WSDM*, 2013.

[95] John Hopcroft, Tiancheng Lou, and Jie Tang. Who will follow you back?: Reciprocal relationship prediction. In *CIKM*, 2011.

[96] Bernardo A Huberman, Daniel M Romero, and Fang Wu. Social networks that matter: Twitter under the microscope. *Available at SSRN 1313405*, 2008.

[97] CJ Hutto, Sarita Yardi, and Eric Gilbert. A longitudinal study of follow predictors on twitter. In *CHI*, 2013.

[98] José Luis Iribarren and Esteban Moro. Affinity paths and information diffusion in social networks. *Social networks*, 2011.

[99] Ratkiewicz Jacob, Conover Michael, Meiss Mark, Gonçalves Bruno, Patil Snehal, Flammini Alessandro, and Menczer Filippo. Truthy: mapping the spread of astroturf in microblog streams. In *WWW companion*, 2011.

[100] Lee Janghyuk, Lee Jong-Ho, and Lee Dongwon. Impacts of tie characteristics on online viral diffusion. *Comm. Assoc. Info. Sys.*, 2009.

[101] Bernard J. Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*, 60(11), 2009.

[102] Nathalie Japkowicz and Mohak Shah. *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, 2011.

[103] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: Understanding microblogging usage and communities. In *WebKDD/SNA Workshop, KDD*, 2007.

[104] Maximilian Jenders, Gjergji Kasneci, and Felix Naumann. Analyzing and predicting viral tweets. In *WWW Companion*, 2013.

[105] An Jisun, Cha Meeyoung, Gummadi P. Krishna, and Crowcroft Jon. Media landscape in twitter: A world of new conventions and political diversity. In *ICWSM*, 2011.

[106] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR*, 2005.

[107] Appelo Jurgen. Twitter top 100 for software developers. In *http://www.noop.nl/2009/02/twitter-top-100-for-software-developers.html*, 2009.

[108] S. Jurvetson. From the ground floor: What exactly is viral marketing? *Red Herring Communications*, 2000.

[109] Krishna Y Kamath and James Caverlee. Spatio-temporal meme prediction: learning what hashtags will be popular where. In *CIKM*, 2013.

[110] Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

[111] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.

[112] Elham Khabiri, James Caverlee, and Krishna Y Kamath. Predicting semantic annotations on the real-time web. In *HT*, 2012.

[113] Myunghwan Kim and Jure Leskovec. Latent multi-group membership graph model. In *ICML*, 2012.

[114] Suin Kim, JinYeong Bak, and Alice Haeyun Oh. Do you feel what i feel? social aspects of emotions in twitter conversations. In *ICWSM*, 2012.

[115] Younghoon Kim and Kyuseok Shim. Twitobi: A recommendation system for twitter using probabilistic modeling. In *ICDM*, 2011.

[116] Funda Kivran-Swaine, Priya Govindan, and Mor Naaman. The impact of network structure on breaking ties in online social networks: Unfollowing on twitter. In *CHI*, 2011.

[117] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of ACM*, 46:604–632, 1999.

[118] Isabel M Kloumann and Jon M Kleinberg. Community membership identification from small seed sets. In *KDD*, 2014.

[119] Farshad Kooti, Haeryun Yang, Meeyoung Cha, P Krishna Gummadi, and Winter A Mason. The emergence of conventions in online social networks. In *ICWSM12*, 2012.

[120] Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. Latent dirichlet allocation for tag recommendation. In *ReSys*, 2009.

[121] Z. Kunda. Motivated inference: Self-serving generation and evaluation of causal theories. *Journal of Personality and Social Psychology*, 53(4):636, 1987.

[122] Haewoon Kwak, Hyunwoo Chun, and Sue Moon. Fragile online relationship: a first look at unfollow dynamics in twitter. In *CHI*, 2011.

[123] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *WWW*, 2010.

[124] Haewoon Kwak, Sue B Moon, and Wonjae Lee. More of a receiver than a giver: Why do people unfollow in twitter? In *ICWSM*, 2012.

[125] Su Mon Kywe, Tuan-Anh Hoang, Ee-Peng Lim, and Feida Zhu. On recommending hashtags in twitter networks. In *SocInfo*. 2012.

[126] Conrad Lee, Aaron McDaid, Fergal Reid, and Neil J Hurley. Detecting highly overlapping community structure by greedy clique expansion. In *SNA workshop, KDD*, 2010.

[127] Kyumin Lee, Jalal Mahmud, Jilin Chen, Michelle Zhou, and Jeffrey Nichols. Who will retweet this? detecting strangers from twitter to retweet information. *TIST*, 2015.

[128] Janette Lehmann, Bruno Gonçalves, José J. Ramasco, and Ciro Cattuto. Dynamical classes of collective attention in twitter. In *WWW*, 2012.

[129] Kristina Lerman and Rumi Ghosh. Information contagion: An empirical study of the spread of news on digg and twitter social networks. *ICWSM*, 10, 2010.

[130] Aaron Q Li, Amr Ahmed, Sujith Ravi, and Alexander J Smola. Reducing the sampling complexity of topic models. In *KDD*, 2014.

[131] Kar Wai Lim and Wray Buntine. Twitter opinion topic model: Extracting product opinions from tweets by leveraging hashtags and sentiment lexicon. In *CIKM*, 2014.

[132] Jimmy Lin and Gilad Mishne. A study of "churn" in tweets and real-time search queries. In *ICWSM*, 2012.

[133] Tianyi Lin, Wentao Tian, Qiaozhu Mei, and Hong Cheng. The dual-sparse topic model: mining focused topics and focused terms in short text. In *WWW*, 2014.

[134] Yu-Ru Lin, Brian Keegan, Drew Margolin, and David Lazer. Rising tides or rising stars?: Dynamics of shared attention on twitter during media events. *PLoS ONE*, 2014.

[135] Yu-Ru Lin, Drew Margolin, Brian Keegan, Andrea Baronchelli, and David Lazer. #bigbirds never die: Understanding social dynamics of emergent hashtags. In *ICWSM*, 2013.

[136] Nelly Litvak, Werner RW Scheinhardt, and Yana Volkovich. In-degree and pagerank: Why do they follow similar power laws? *Internet mathematics*.

[137] Guannan Liu, Yanjie Fu, Tong Xu, Hui Xiong, and Guoqing Chen. Discovering temporal retweeting patterns for social media marketing campaigns. In *ICDM*, 2014.

[138] Guannan Liu, Yanjie Fu, Tong Xu, Hui Xiong, and Guoqing Chen. Discovering temporal retweeting patterns for social media marketing campaigns. In *ICDM*, 2014.

[139] Jun S Liu. The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem. *J. Amer. Stat. Assoc*, 1994.

[140] Lu Liu, Jie Tang, Jiawei Han, Meng Jiang, and Shiqiang Yang. Mining topic-level influence in heterogeneous networks. In *CIKM*, 2010.

[141] Yan Liu, Alexandru Niculescu-Mizil, and Wojciech Gryc. Topic-link lda: joint models of topic and author community. In *ICML*. ACM, 2009.

[142] M. Lodge and C. Taber. Three steps toward a theory of motivated political reasoning. In *Elements of Political Reason*. Cambridge University Press, London, 2000.

[143] Tiancheng Lou, Jie Tang, John Hopcroft, Zhanpeng Fang, and Xiaowen Ding. Learning to predict reciprocity and triadic closure in social networks. *TKDD*, 7(2):5, 2013.

[144] Zhunchen Luo, Miles Osborne, Jintao Tang, and Ting Wang. Who will retweet me? finding retweeters in twitter. In *SIGIR*, 2013.

[145] Zhiqiang Ma, Wenwen Dou, Xiaoyu Wang, and Srinivas Akella. Tag-latent dirichlet allocation: Understanding hashtags and their relationships. In *WI/IAT 2013*, 2013.

[146] Zongyang Ma, Aixin Sun, and Gao Cong. Will this# hashtag be popular tomorrow? In *SIGIR*, 2012.

[147] Zongyang Ma, Aixin Sun, Quan Yuan, and Gao Cong. Tagging your tweets: A probabilistic modeling of hashtag annotation in twitter. In *CIKM*, 2014.

[148] Sofus A Macskassy. On the study of social interactions in twitter. In *ICWSM*, 2012.

[149] Sofus A Macskassy and Matthew Michelson. Why do people retweet? anti-homophily wins the day! In *ICWSM*, 2011.

[150] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. *Introduction to Information Retrieval.* Cambridge University Press, 2008.

[151] Andrew McCallum, Xuerui Wang, and Andrés Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research*, pages 249–272, 2007.

[152] Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *SIGIR*, 2013.

[153] Marcelo Mendoza, Barbara Poblete, and Carlos Castillo. Twitter under crisis: Can we trust what we rt? In *Proceedings of the first workshop on social media analytics*, pages 71–79. ACM, 2010.

[154] Matthew Michelson and Sofus A. Macskassy. Discovering users' topics of interest on twitter: A first look. In *AND Workshop, CIKM*, 2010.

[155] Seth A Myers and Jure Leskovec. The bursty dynamics of the twitter information network. In *WWW*, 2014.

[156] Seth A Myers, Chenguang Zhu, and Jure Leskovec. Information diffusion and external influence in networks. In *KDD*, 2012.

[157] Meenakshi Nagarajan, Hemant Purohit, and Amit P Sheth. A qualitative examination of topical tweet and retweet practices. *ICWSM*, 2010.

[158] Ramesh M. Nallapati, Amr Ahmed, Eric P. Xing, and William W. Cohen. Joint latent topic models for text and citations. In *KDD*, 2008.

[159] David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. Distributed algorithms for topic models. *The Journal of Machine Learning Research*, 10:1801–1828, 2009.

[160] M. E. J. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA*, pages 8577–8582, 2006.

[161] Mark EJ Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133, 2004.

[162] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.

[163] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[164] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[165] Ye Pan, Feng Cong, Kailong Chen, and Yong Yu. Diffusion-aware personalized social update recommendation. In *RecSys*, 2013.

[166] Adam L. Penenberg. *Viral loop : from Facebook to Twitter, how today's smartest businesses grow themselves*. Hyperion, 2009.

[167] Huan-Kai Peng, Jiang Zhu, Dongzhen Piao, Rong Yan, and Ying Zhang. Retweet modeling using conditional random fields. In *workshop, ICDM*, 2011.

[168] Marco Pennacchiotti and Ana-Maria Popescu. Democrats, republicans and starbucks afficionados: user classification in twitter. In *KDD*, 2011.

[169] Marco Pennacchiotti and Ana-Maria Popescu. Democrats, republicans and starbucks afficionados: user classification in twitter. In *KDD*, 2011.

[170] Renana Peres, Eitan Muller, and Vijay Mahajan. Innovation diffusion and new product growth models: A critical review and research directions. *International Journal of Research in Marketing*, 27(2):91–106, 2010.

[171] Sasa Petrovi, Miles Osborne, and Victor Lavrenko. Rt to win! predicting message propagation in twitter. In *ICWSM*, 2011.

[172] René Pfitzner, Antonios Garas, and Frank Schweitzer. Emotional divergence influences information spreading in twitter. In *ICWSM*, 2012.

[173] Doug Pierce, David P. Redlawsk, William W. Cohen, Tae Yano, and Ramnath Balasubramanyan. Social and affective responses to political information. *APSA*, 2012.

[174] Deborah A Prentice, Dale T Miller, and Jenifer R Lightdale. Asymmetries in attachments to groups and to their members: Distinguishing between common-identity and common-bond groups. *Key Readings in Social Psychology*, page 83, 1994.

[175] Ioannis Psorakis, Stephen Roberts, Mark Ebden, and Ben Sheldon. Overlapping community detection using bayesian non-negative matrix factorization. *Physical Review E*, 83(6):066114, 2011.

[176] Hemant Purohit, Andrew Hampton, Valerie L Shalin, Amit P Sheth, John Flach, and Shreyansh Bhatt. What kind of# conversation is twitter? mining# psycholinguistic cues for emergency coordination. *Computers in Human Behavior*, 29(6):2438–2447, 2013.

[177] Minghui Qiu, Jing Jiang, and Feida Zhu. It is not just what we say, but how we say them: Lda-based behavior-topic model. In *SDM*, 2013.

[178] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.

[179] Daniel Ramage, Susan T Dumais, and Daniel J Liebling. Characterizing microblogs with topic models. In *ICWSM*, 2010.

[180] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *ECML*, 2009.

[181] Jacob Ratkiewicz, Michael Conover, Mark Meiss, Bruno Gonalves, Alessandro Flammini, and Filippo Menczer. Detecting and tracking political abuse in social media. In *ICWSM*, 2011.

[182] D. P. Redlawsk, A. J. W. Civettini, and K. M. Emmerson. The affective tipping point: Do motivated reasoners ever get it? *Political Psychology*, 31(4):563–593, 2010.

[183] David Redlawsk. Hot cognition or cool consideration? testing the effects of motivated reasoning on political decision making. *The Journal of Politics*, 64(04):1021–1044, 2002.

[184] David Redlawsk. Motivated reasoning, affect, and the role of memory in voter decision-making. In David Redlawsk, editor, *Feeling Politics: Emotion in Political Information Processing*. Palgrave Macmillan, 2006.

[185] Alan Ritter, Colin Cherry, and Bill Dolan. Unsupervised modeling of twitter conversations. In *NAACL HLT*, 2010.

[186] E. M. Rogers. *Diffusion of Innovation*. The Free Press, New Yory, 1962.

[187] Daniel Romero, Wojciech Galuba, Sitaram Asur, and Bernardo Huberman. Influence and passivity in social media. In *Machine learning and knowledge discovery in databases*. 2011.

[188] Daniel M. Romero, Brendan Meeder, and Jon Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In *WWW*, 2011.

[189] Daniel Mauricio Romero and Jon M Kleinberg. The directed closure process in hybrid social-information networks, with an analysis of link formation on twitter. In *ICWSM*, 2010.

[190] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *UAI*, 2004.

[191] Sheldon M. Ross. *Introduction to Probability Models, Ninth Edition*. Academic Press, Inc., Orlando, FL, USA, 2006.

[192] Jianhua Ruan and Weixiong Zhang. An efficient spectral algorithm for network community discovery and its applications to biological and social networks. In *ICDM*, 2007.

[193] Yiye Ruan, David Fuhry, and Srinivasan Parthasarathy. Efficient community detection in large networks using content and links. In *WWW*, 2013.

[194] Mrinmaya Sachan, Danish Contractor, Tanveer A. Faruquie, and L. Venkata Subramaniam. Using content and interactions for discovering communities in social networks. In *WWW*, 2012.

[195] Mrinmaya Sachan, Avinava Dubey, Shashank Srivastava, Eric P. Xing, and Eduard Hovy. Spatial compactness meets topical consistency: Jointly modeling links and content for community detection. In *WSDM*, 2014.

[196] Diego Saez-Trumper, Giovanni Comarela, Virgílio Almeida, Ricardo Baeza-Yates, and Fabrício Benevenuto. Finding trendsetters in information networks. In *KDD*, 2012.

[197] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW*, 2010.

[198] Johannes Schantl, Rene Kaiser, Claudia Wagner, and Markus Strohmaier. The utility of social and topical factors in anticipating repliers in twitter conversations. In *WebSci*, 2013.

[199] David A. Shamma, Jude Yew, Lyndon Kennedy, and Elizabeth F. Churchill. Viral actions: Predicting video view counts using synchronous sharing behaviors. In *ICWSM*, 2011.

[200] Huawei Shen, Xueqi Cheng, Kai Cai, and Mao-Bin Hu. Detect overlapping and hierarchical community structure in networks. *Physica A: Statistical Mechanics and its Applications*, 388(8):1706–1712, 2009.

[201] Daniel Sousa, Luís Sarmento, and Eduarda Mendes Rodrigues. Characterization of the twitter @replies network: Are user ties social or topical? In *SMUC workshop, CIKM*, 2010.

[202] Kate Starbird and Leysia Palen. (how) will the revolution be retweeted?: information diffusion and the 2011 egyptian uprising. In *CSCW*, 2012.

[203] S. Stieglitz and Linh Dang-Xuan. Political communication and influence through microblogging–an empirical analysis of sentiment in twitter messages and retweet behavior. In *HICSS*, 2012.

[204] Bongwon Suh, Lichan Hong, Peter Pirolli, and Ed H. Chi. Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In *SocialCom*, 2010.

[205] Tao Sun, Ming Zhang, and Qiaozhu Mei. Unexpected relevance: An empirical study of serendipity in retweets. In *ICWSM*, 2013.

[206] Gabor Szabo and Bernardo A. Huberman. Predicting the popularity of online content. *Commun. ACM*, 2010.

[207] Partha Pratim Talukdar and Fernando Pereira. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. ACL, 2010.

[208] Chenhao Tan, Lillian Lee, and Bo Pang. The effect of wording on message propagation: Topic-and author-controlled natural experiments on twitter. In *ACL*, 2014.

[209] Shulong Tan, Yang Li, Huan Sun, Ziyu Guan, Xifeng Yan, Jiajun Bu, Chun Chen, and Xiaofei He. Interpreting the public sentiment variations on twitter. *IEEE TKDE*, 2014.

[210] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *KDD*, 2008.

[211] Rosemary Thackeray, Scott H Burton, Christophe Giraud-Carrier, Stephen Rollins, and Catherine R Draper. Using twitter for breast cancer prevention: an analysis of breast cancer awareness month. *BMC cancer*, 2013.

[212] Oren Tsur and Ari Rappoport. What's in a hashtag?: Content based prediction of the spread of ideas in microblogging communities. In *WSDM*, 2012.

[213] Andranik Tumasjan, Timm Oliver Sprenger, Philipp G. Sandner, and Isabell M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *ICWSM*, 2010.

[214] Toma Turk and Peter Trkman. Bass model estimates for broadband diffusion in european countries. *Technological Forecasting and Social Change*, 79(1):85 – 96, 2012.

[215] Johan Ugander and Lars Backstrom. Balanced label propagation for partitioning massive graphs. In *WSDM*, 2013.

[216] Ibrahim Uysal and W. Bruce Croft. User oriented tweet ranking: A filtering approach to microblogs. In *CIKM*, 2011.

[217] Sarah Vieweg, Amanda L. Hughes, Kate Starbird, and Leysia Palen. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *CHI*, 2010.

[218] Jan Vosecky, Di Jiang, Kenneth Wai-Ting Leung, and Wilfred Ng. Dynamic multi-faceted topic discovery in twitter. In *CIKM*, 2013.

[219] Jan Vosecky, Di Jiang, Kenneth Wai-Ting Leung, Kai Xing, and Wilfred Ng. Integrating social and auxiliary semantics for multifaceted topic modeling in twitter. *TOIT*, 2014.

[220] Alastair J. Walker. An efficient method for generating discrete random variables with general distributions. *ACM Trans. Math. Softw.*, 1977.

[221] Aobo Wang, Tao Chen, and Min-Yen Kan. Re-tweeting from a linguistic perspective. In *NAACL-HLT 2012 Workshop on Language in Social Media*, 2012.

[222] Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In *KDD '11*, 2011.

[223] Fei Wang, Tao Li, Xin Wang, Shenghuo Zhu, and Chris Ding. Community discovery using nonnegative matrix factorization. *Data Mining and Knowledge Discovery*, 2011.

[224] Jinpeng Wang, Wayne Xin Zhao, Yulan He, and Xiaoming Li. Infer user interests via link structure regularization. *TIST*, 2014.

[225] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: finding topic-sensitive influential twitterers. In *WSDM*, 2010.

[226] Lilian Weng, Filippo Menczer, and Yong-Yeol Ahn. Virality prediction and community structure in social networks. *Scientific reports*, 2013.

[227] Scott White and Padhraic Smyth. A spectral clustering approach to finding communities in graph. In *SDM*, 2005.

[228] Shaomei Wu, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. Who says what to whom on twitter. In *WWW*, 2011.

[229] Pengtao Xie and Eric P Xing. Integrating document clustering and topic modeling. In *UAI*, 2013.

[230] Bo Xu, Yun Huang, Haewoon Kwak, and Noshir Contractor. Structures of broken ties: Exploring unfollow behavior on twitter. In *CSCW*, 2013.

[231] Zhiheng Xu and Qing Yang. Analyzing user retweet behavior on twitter. In *ASONAM*, 2012.

[232] Zhiqiang Xu, Yiping Ke, Yi Wang, Hong Cheng, and James Cheng. A model-based approach to attributed graph clustering. In *SIGMOD*, 2012.

[233] Rui Yan, Mirella Lapata, and Xiaoming Li. Tweet recommendation with graph co-ranking. In *ACL*, 2012.

[234] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. A biterm topic model for short texts. In *WWW*, 2013.

[235] Jaewon Yang and Jure Leskovec. Modeling information diffusion in implicit networks. In *ICDM*, 2010.

[236] Jaewon Yang and Jure Leskovec. Patterns of temporal variation in online media. In *WSDM*, 2011.

[237] Jaewon Yang and Jure Leskovec. Community-affiliation graph model for overlapping network community detection. In *ICDM*, 2012.

[238] Jaewon Yang, Julian McAuley, and Jure Leskovec. Community detection in networks with node attributes. In *ICDM*, 2013.

[239] Jaewon Yang, Julian McAuley, and Jure Leskovec. Detecting cohesive and 2-mode communities indirected and undirected networks. In *WSDM*, 2014.

[240] Jiang Yang and Scott Counts. Predicting the speed, scale, and range of information diffusion in twitter. In *ICWSM*, 2010.

[241] Lei Yang, Tao Sun, Ming Zhang, and Qiaozhu Mei. We know what @you #tag: Does the dual role affect hashtag adoption? In *WWW*, 2012.

[242] Shuang-Hong Yang, Alek Kolcz, Andy Schlaikjer, and Pankaj Gupta. Large-scale high-precision topic modeling on twitter. In *KDD*, 2014.

[243] Shuang-Hong Yang, Alek Kolcz, Andy Schlaikjer, and Pankaj Gupta. Large-scale high-precision topic modeling on twitter. In *KDD*, 2014.

[244] Yang Yang, Jie Tang, Cane Wing-ki Leung, Yizhou Sun, Qicong Chen, Juanzi Li, and Qiang Yang. Rain: Social role-aware information diffusion. *AAAI*, 2014.

[245] Zi Yang, Jingyi Guo, Keke Cai, Jie Tang, Juanzi Li, Li Zhang, and Zhong Su. Understanding retweeting behaviors in social networks. In *CIKM*, 2010.

[246] Tae Yano, William W. Cohen, and Noah A. Smith. Predicting response to political blog posts with topic models. In *NAACL*, 2009.

[247] Dawei Yin, Liangjie Hong, and Brian D. Davison. Structural link analysis and prediction in microblogs. In *CIKM*, 2011.

[248] Zhijun Yin, Liangliang Cao, Quanquan Gu, and Jiawei Han. Latent community topic analysis: Integration of community discovery with topic modeling. *ACM Trans. Intell. Syst. Technol.*, 2012.

[249] Jinhui Yuan, Fei Gao, Qirong Ho, Wei Dai, Jinliang Wei, Xun Zheng, Eric Po Xing, Tie-Yan Liu, and Wei-Ying Ma. Lightlda: Big topic models on modest computer clusters. In *WWW*, 2015.

[250] Eva Zangerle and Wolfgang Gassler. Recommending #-tags in twitter. In *Workshop on Semantic Adaptive Social Web, UMAP*, 2011.

[251] Fabio Massimo Zanzotto, Marco Pennacchiotti, and Kostas Tsioutsiouliklis. Linguistic redundancy in twitter. In *EMNLP*, 2011.

[252] Michele Zappavigna. Ambient affiliation: A linguistic perspective on twitter. *New Media & Society*, 13(5), 2011.

[253] E. Zeidler. *Applied functional analysis: applications to mathematical physics*. Springer-Verlag, 1995.

[254] Jing Zhang, Biao Liu, Jie Tang, Ting Chen, and Juanzi Li. Social influence locality for modeling retweeting behaviors. In *IJCAI*, 2013.

[255] Jing Zhang, Jie Tang, Juanzi Li, Yang Liu, and Chunxiao Xing. Who influenced you? predicting retweet via social influence locality. *TKDD*, 2015.

[256] Qi Zhang, Yeyun Gong, Ya Guo, and Xuanjing Huang. Retweet behavior prediction using hierarchical dirichlet process. In *AAAI*, 2015.

[257] Dejin Zhao and Mary Beth Rosson. How and why people twitter: The role that micro-blogging plays in informal communication at work. In *GROUP*, 2009.

[258] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *ECIR*, 2011.

[259] Liu Zhiming, Liu Lu, and Li Hong. Determinants of information retweeting in microblogging. *Internet Research*, 2012.

[260] Ding Zhou, Eren Manavoglu, Jia Li, C. Lee Giles, and Hongyuan Zha. Probabilistic models for discovering e-communities. In *WWW*, 2006.

[261] Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. Graph clustering based on structural/attribute similarities. *VLDB Endowment*, 2(1):718–729, 2009.

[262] Zicong Zhou, Roja Bandari, Joseph Kong, Hai Qian, and Vwani Roychowdhury. Information resonance on twitter: watching iran. In *SOMA workshop, KDD*, 2010.