ABSTRACT

Title of Dissertation:	DESCRIBING URGENT EVENT DIFFUSION ON TWITTER USING NETWORK STATISTICS
	Hechao Sun, Doctor of Philosophy, 2017
Dissertation directed by:	Assistant Professor, Bill Rand, Department of Business Management, NC State University

In this dissertation, I develop a novel framework to study the diffusion of urgent events through the popular social media platform—Twitter. Based on my literature review, this is the first comprehensive study on urgent event diffusion through Twitter. I observe similar diffusion patterns among different data sets and adopt the "cross prediction" mode to handle the early time prediction problem. I show that the statistics from the network of Twitter retweets can not only provide profound insights about event diffusion, but also can be used to effectively predict user influence and topic popularity. The above findings are consistent across various experiment settings. I also demonstrate that linear models consistently outperform state-of-art nonlinear ones in both user and hashtag prediction tasks, possibly implying the strong log-linear relationship between selected prediction features and the responses, which potentially could be a general phenomenon in the case of urgent event diffusion.

DESCRIBING URGENT EVENT DIFFUSION ON TWITTER USING NETWORK STATISTICS

by

Hechao Sun

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2017

Advisory Committee:
Professor William Rand, Chair
Professor Michelle Girvan
Professor Jeffrey Herrmann
Professor Shawn Mankad
Professor Paul Smith

Dedication

This thesis is dedicated to everyone who has helped and supported me during my whole degree.

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor Dr. Bill Rand for the continuous support of my Ph.D. study and research, for his patience, motivation, enthusiasm, immense knowledge and expertise. His guidance helped me through the research and the writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Shawn Mankad, Dr. Paul Smith, Dr. Jeffrey Herrmann, and Dr. Michelle Girvan, for their insightful comments and encouragement, but also for the hard questions that incentivized me to broaden my research from various perspectives.

Also I would like to express my heartfelt thanks to my parents who have supported me spiritually throughout the writing process and throughout my life in general.

Last, but not least, I want to thank all of the colleagues in my research group who gave me suggestions and enlightened my research, including but not limited to: Dr. David Darmon, Dr. Keith Burghardt, Mr. Arjuna Ariyaratne, and Mr. Chen Wang.

Table of Contents

List of Tables

Table **Page**

List of Figures

Figure **Page**

List of Abbreviations

Abbreviations	**Full name**
B	Baseline features
BSD	Baseline+static network+dynamic network features
CDF	Cumulative distribution function
COO	Co-occurrence
CV	Cross validation
DL	Deep learning
F-F network	Following follower network
GLM	Generalized linear model
LDA	Latent Dirichlet Allocation
LSCC	Largest strongly connected component
LWCC	Largest weakly connected component
RF	Random forest
RMSE	Root mean square error
RT	Retweet
SD	Static network+dynamic network features

Chapter 1: Introduction

1.1 Background

1.1.1 Motivation

In recent years, the emergence of popular social media platforms, such as Facebook and Twitter, has largely changed the way people communicate and interact with each other. These online social networks, not only provide global platforms for users around the world to share information and express opinions, but also generate massive amounts of unstructured data resulting from the human-to-human interactions. These huge amounts of data are still far from been fully utilized, which, at present, creates great challenges for people or organizations to use the data to make better and intelligent decisions (Evans, 2010). One crucial area still to be explored is the study of information diffusion (spreading) on these platforms, which has great business potential and marketing implications (Evans, 2010) (Chen, 2010) (Stieglitz, 2013)

Information diffusion can be considered a subtopic of the study of general diffusion, and it includes a broad range of research directions. It can be classified into different sub-categories depending on the research context—exploratory (Zhou, 2011) vs. predictive analysis (Kupavskii, 2012), parametric (with specific concrete diffusion models) (Galuba, 2010) vs. non-parametric (empirical) studies (Lerman, 2010), diffusion mechanism study (Romero, 2011) vs. user influence study (Bakshy, 2011) or topic popularity study (Ma, 2013) etc. As to the different platforms themselves—whether they are large social media platforms or only restricted small social networks, it is not surprising to find that many research results are generalizable and can be directly applied to one another (Borge-Holthoefer, 2013). This is because although these platforms do possess diverse properties, they share much more in common—which can be summarized as general principles of information diffusion (more details in the next section).

For my work, I concentrate on one small aspect of this large topic—information diffusion study during urgent events on the well-known social media platform—Twitter. Here, urgent events are defined as any large event that results in rapid and large-scale diffusion. More specifically it may include the following cases:

1. A large emergency or crisis—such as natural disasters, terrorist attack, or any other types of events that will cause large-scale public concern.

2. General world-wide breaking news that attracts great public interest.

3. Other types of explosive information that can spread over with comparable scale and speed, such as diffusion of news about notable brands.

Although in my study I mainly use the data from the first category, I believe the results should hold consistently for other types of urgent events. This is because as stated above, all these urgent events share common diffusion patterns—vast spreading speed and large spreading scales regardless of the types of urgent event. I hypothesize that all the tools I have developed can be easily migrated.

However those special properties of urgent events diffusion will add significant difficulty on the predictive analysis of the data, since an early period prediction will be necessary for stake holders to respond in time and to make proper decisions based on the prediction results. This is the main emphasis of my work, which includes prediction on user influence and topic popularity during the urgent diffusion.

1.1.2 Why retweets on Twitter

Being a microblogging service with over 300 million monthly active users, Twitter is commonly used to propagate information using short text messages. Though the conversations occurred online, they can provide deep insights in how generally people behave and offer enough information to understand human behavior across a variety of fields (Bollen, 2011) (Borondo, 2012). Especially for urgent events or crises, Twitter has become a powerful medium that propagates news globally at a high rate. Therefore, I use information diffusion on Twitter as my study case to give practical implications and to develop reasonable decisions or strategies.

Engagement	Functionality	Network
Follow others	Build friendship	Following-follower network
Create new tweets	Start new conversation	No network directly formed
Mention others in tweets	Involve in conversation	Mention network
Retweet others' tweets	Spread information	Retweet network

Table 1.1 Twitter engagement and network formation

Table 1.1 gives different types of engagement that a user can participate in on Twitter. Posting tweets is the content generating step, but how much the content will be exposed will depend on the following-follower network (F-F network in short for simplicity) and the privacy settings (a user can control whether his or her tweets can be seen by the general public). This is why study of the F-F network has been the mainstream research area (Kwak, 2010) (Galuba, 2010): it more or less controls the diffusion potentials of new content, in most situations mentioning and retweeting can only occur on the backbone of the F-F network (Figure 1.1). But this can also be the reason why study of F-F network alone is not enough—it just provides the backbone for the diffusion not the actual diffusion pathways. Moreover, research shows that using number of followers as the user influence measure could be misleading (Cha, 2010) , where large divergence has been observed between user influences measured number of followers and number of retweets, and the latter measure is usually more meaningful in practical applications.

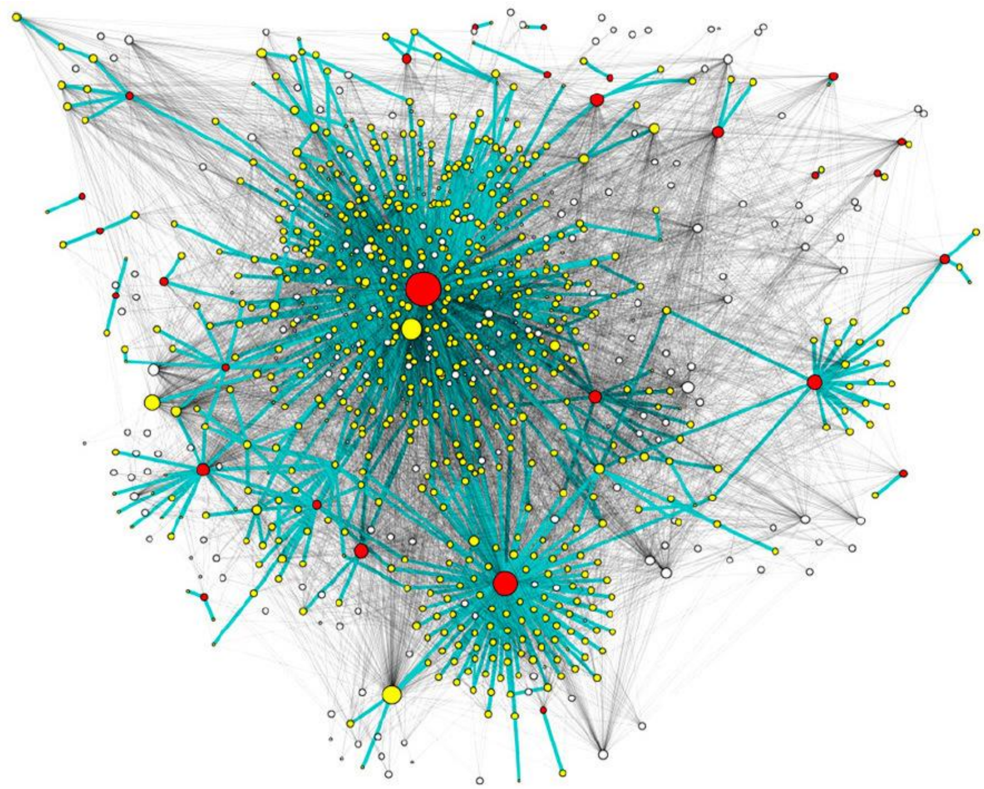


Figure 1.1 Information diffusion by retweets on backbone of F-F network (Morales, 2014)

As to retweeting, it is a typical Twitter behavior where users can share tweets that they believe are interesting or important to their followers or general public (Table 1.1). Under the context when real life events occur, this behavior can serve to spread important information through the whole Twitter network (Kwak, 2010).

However, retweeting is not the only way to diffuse important information on Twitter—other engagement behavior such as tweets with links or hashtags and mentions are also popular alternatives for retweeting (Galuba, 2010) (Bakshy, 2011) (Rattanaritnont, 2012). But compared to mentions, URLs and hashtag tweets, retweets have the following distinct properties:

1. Retweets can be considered as the minimum pathways for information diffusion. Perhaps not all informative tweets are retweeted by the users, but people choose to

retweet usually when they believe the focal tweet is more important and informative. This argument seems to be how some tweets gain a high number of retweets. Thus retweets should highlight tweets which are crucial in information spreading. Using retweets is a simple and straightforward way to capture the underlying diffusion pathways in the lower bound sense.

2. Retweets can spread information regardless of the origin—they are not restricted by the forms of the tweets. Tweets that include URLs, hashtags or simply plain text are all able to deliver useful information as long as they are generating retweets. Though from my data I observe people do use more URLs and hashtags during urgent diffusion, there are still many examples of informative tweets—which got a lot of retweets and do not include URLs or hashtags: for example, tweets like "HURRICANE WATCH for CT shoreline!!!! Just issued!" occur in the hurricane data sets. On the other hand, hashtags or URLs can often be the breeding ground for all types of spam messages (Grier, 2010) (Thomas, 2011)

3. Another advantage of using retweets compared to tweets with URLs or hashtags is that retweets can provide solidly tractable pathways for information diffusion models (Zhou, 2011) (Hong, 2011) (Kupavskii, 2012). While for URLs and hashtags tweets, an actual diffusion model that defines the information pathways is needed to do the analysis—the diffusion models defined on chronological adoption of URLs for example (Bakshy, 2011). Also as mentioned in the same literature, there are situations where chronological adoption of URLs or hashtags does not imply influence at all--when users could just cite them independently, while in contrast retweets with URLs or hashtags can be considered as stronger indicators of influence than these merely chronological influence assignments. Moreover, the users' detailed following-follower structure is often necessary for these influence models definition, which is generally not available especially for large number of users.

4. Compared to the other alternative diffusion pathways—mentions, retweets are considered as a subset of them in the Twitter context. Thus mentions still possess those advantages over URLs and hashtags, but the problem with mentions is that they also include casual mutual conversations among users, which cannot be accounted as really informative tweets.

Because of these reasons, in my study I focus on retweets for the information diffusion study. I understand that they may not reflect the actual diffusion pathways, i.e., since users can communicate outside the Twitter platform. Actually I am seldom fully exposed to the ground truth information diffusion pathways unless I am keep tracking of all users and all of their communication, which is impossible for big urgent event diffusion on Twitter—a world-wide information exchange platform. Therefore I believe using retweets is the currently best available choice to keep track of and study the information diffusion on Twitter.

1.2 Related work

1.2.1 Information diffusion study

Studies concerning information diffusion in general can be divided into descriptive analysis and predictive analysis, and information diffusion modeling usually takes both aspects into consideration. My work also contains both descriptive and predictive analysis during urgent events diffusion.

In theory, information generally propagates in the form of information cascades through various networks. According to Bikhchandani's defination, the propagation results from actions of individuals to follow behavior of preceding individuals without regarding to their own information (Bikhchandani, 1992). Different types of information cascades occur under different networks (Zhou, 2011) (Borge-Holthoefer, 2013) (Cheng, 2014). As to Twitter particularly, the information cascades are usually the retweet cascades or retweet trees (Kwak, 2010) (Zhou, 2011), though there are also more complex definitions (Wu, 2011) (Hui, 2012) (Taxidou, 2014).

The descriptive studies of various networks--blog graph (Gruhl, 2004), Facebook photo sharing graph (Cheng, 2014), Twitter retweet graph (Borondo, 2012) and even more general social networks (Borge-Holthoefer, 2013) have shown that information cascades share common properties: one property is the broad but very skewed size distribution, which usually resembles the power law distribution; the other one is the relative shallow depth and tree-like topology. With these resemblances under various contexts, the methods can usually be used interchangeably regardless of the underlying networks and diffusion models. Previous research explored information diffusion using various models: an epidemic modeling approach is the classic way to study the general diffusion process and includes diffusions other than information diffusion (Gruhl, 2004) (Khelil, 2002); there are also other nonparametric influence models such as the threshold model (Karimi, 2013) and the linear influence model (Yang J. &., 2010). However, many of these diffusion models require knowledge of the underlying affinity network—in the Twitter case it would be the Following-Follower (F-F) network, which provides the backbone for potential diffusion to occur and is not available in the data sets used for my prediction study (more details in data description). Yang (Yang J. &., 2010) built an influence model without a prior knowledge of the F-F network and was able to make good predictions about influence. But his approach does not fit well into my prediction task because i) He studied diffusion of various hashtags instead of diffusion of a specific event; ii) He used data sets that span a long period of time but I only have limited data information for the prediction since it is based on an urgent event diffusion.

Despite of the model differences, all the studies more or less use the temporal properties of diffusion—e.g. the dynamics of the associated information cascades. Given the context of my problem, a very relevant work on information diffusion study is from Rogers (Rogers, 2014). She did the diffusion study through exploring properties of the retweet chains—e.g. dynamics and topology. Moreover she used the same 15k_user data sets (more details later) as I do for the diffusion tracking study. But she did not perform any prediction analysis and she did not use the underlying retweet network properties to characterize the diffusion process, which should have both high explanatory power and high predictive power. I attempt to utilize all available network properties (more details in later section) to reveal elaborate dynamics of the diffusion that are not easily exposed using previous approaches. Thus the diffusion study in this dissertation can be considered as study of temporal networks (Holme, 2012), and I adopt the approach "representing temporal data as a static graph" mentioned in the paper.

1.2.2 Prediction during urgent events diffusion

As to urgent events, crises or emergencies, Palen and colleagues developed comprehensive studies (Hughes, 2009) (Vieweg, 2010). But the majority of his work focuses on descriptive analysis—information extraction, social impact and user behavior etc. Due to the high spread rate and large spread scale of information during urgent events, prediction is a challenging task, but at the same time it also demonstrates great application potentials—e.g. providing recommendations to stake holders who are interested in the urgent diffusion process. Though little previous work exists on urgent event diffusion prediction, I can still draw useful lessons from previous related work, and then make proper accommodations. As noted in the first section, the prediction tasks include user influence prediction and topic popularity prediction. However, based on both previous literatures and my discoveries they are very closely related from each other (especially when the topics are referred to as hashtags in Twitter), many of the models and frameworks can actually be used interchangeably.

User influence characterization and prediction has always been one of the central topics in the field of general diffusion study—not just restricted to information diffusion. Strict definitions of diffusion context and influence are the first things to consider in developing any study on user influence. Since there is always a certain underlying network that can have the diffusion procedure run through, and nodes within the network will represent the actual users, thus various network centrality scores from that network can be directly used as the user influence measure.

Since the definition and quantification of user influence is contextually subjective, I have to settle a ground truth measure for my prediction study. I use the size of information cascades—the retweet cascades to be the golden-rule measure, which is considered as a standard empirical influence metric for information seed users (Kleinberg, 1999) (Kupavskii, 2012). My fundamental question is which centrality metrics have good prediction power for this influence measure. Various classical network centrality measures (degree, PageRank, HITS scores etc.) (Kwak, 2010) (Cha, 2010) (Galuba, 2010) (Rattanaritnont, 2012) , or additional derived measures (Benzi, 2013) (Laflin, 2013) (Mantzaris, 2013) could be used as the potential predictors. However, there is no comprehensive study to predict node influence using those different network centralities features derived from the underlying diffusion network—the retweet network in my case (more details in next section). I believe these various centrality scores derived from the diffusion network should be closely correlated with users' retweet cascades size and thus serve as potential effective predictors.

Previous research has shown prediction concerning information cascades. For this work, I just pick several examples for illustration and comparison purposes. Bakshy (Bakshy, 2011) created an overall prediction on average size of cascades formed by root users using a decision tree, but the prediction performance is relatively poor due to the highly skew distribution of cascade size, high variability among users behavior, according to the author. Noticing this fact, I do not plan to make general prediction for all nodes, but only focus on "important" ones. This is a practical consideration since the majority of nodes involved in the network will be inactive in the future and I only need to keep track of those really influential ones. My prediction framework starts with this framework as a foundation, and then implements new network features in addition to those have been adopted in the paper. Kupavskii (Kupavskii, 2012) did a thorough search of possible features and obtained decent results on retweet cascades prediction, which provides a broad list of potential features to be used. But what he was trying to predict are cascades size formed by specific tweets, not by users, thus what he is actually doing is predicting tweet influence instead of user influence. Thus, some features he employed—especially the tweet content features, are not usable in my case.

Cheng (Cheng, 2014) avoided direct prediction about the cascade size and innovatively switch the problem to examine whether each cascade will grow over the medium size or double its current size. His work provides insights by discovering significant features in controlling growth of cascades, which is also enlightening in identifying potential influential users who will double their scores in the future. But my main goal is not on the predictability of cascade growth and how the effects of various features change with evolution of cascades. It is less relevant for me to know the detailed growth procedure or whether a cascade will double its size in the future, since the majority of cascades turn out to be small, I only care about nodes that are going to be "important" in the near future—top tiers nodes in the influence scores rank. Also I am studying Twitter retweet network instead of the Facebook photo re-sharing network, thus some features are not applicable to my study. Hofman and colleagues (Martin, 2016) (Hofman, 2017) had a thorough discussion on the predictability of the problem. He mentioned that there was unavoidable inherent variability within the prediction problem even perfect information was given. He suggested unifying the standard for the prediction and evaluation. He also stressed that prediction and interpretation were both important and should be viewed as complements to each other. While his work does have profound implications, it was of less concern to my goal, which was to find the best available solution for the prediction problem given the limited information.

Simmie (Simmie, 2014) applied a hidden Markov model to rank user influence, where he combined information from users' social context--following-follower network information and direct tweets observation together. However, his model is not suitable for my problem because it requires full social link structure of all users—which is not available in my data sets. And it is challenging to collect this information in urgent diffusion case even if I acquire the ability to do so—the diffusion spreads rapidly and the corresponding users graph grows exponentially. One way out would be to narrow the scope only to a much smaller group of users, but the problem is how to select those users and collect their full social links with only limited information given due to the urgent diffusion reality, let alone the fact that the user influence can vary much over time—meaning users selected before may not represent the true influential users in the population. Moreover, estimation or selection of parameters involved in the hidden Markov model requires a large amount of data to obtain meaningful results, which, again, is not feasible in my study. Finally, as the author noted, the scalability would be an issue. Work done by Galuba (Galuba, 2010) illustrates the classical parametric modeling approach using popular epidemic models characterize cascades and making prediction about next hop probability. But again it required information from all users' follower structure. Significant work has been done on predicting the tweet popularity—number of retweets a tweet can get (Hong, 2011) (Jenders, 2013) (Petrovic, 2011). Although these studies have profound pioneering implications, they are not about predicting influential "nodes". And they put much value on content features, which are features derived from single tweets and are difficult to aggregate onto the user level, thus makes them not feasible for user the prediction.

Topic prediction is another area that is well researched. The topic is closely related to the content of the diffusion and refers to be a relative stable and consistent collection of relevant content (Lambrecht, 1996) (e.g. tweets in Twitter), thus topic study is usually involved in the field where users are spreading collections of content or information that can be characterized and properly clustered. Therefore under the context of information diffusion, Twitter is the ideal platform to perform topic analysis with the actual text in the tweets serving as the content to be grouped.

In the Twitter context, one straightforward topic definition is simply using the hashtag (see https://support.twitter.com/articles/49309 for hashtag usage recommendation from Twitter): which should include the keywords or summary of the corresponding tweets, thus can be used to categorize tweets and describe a topic or theme. There have been quite a few successful studies about predicting popularity of hashtags. One very well cited work is from Tsur (Tsur, 2012), where the author predicted weekly hashtags volume using LASSO linear regression. The author adopted a mixture of wide range of features including hashtag content, hashtag tweets content, user F-F network features and some basic temporal features. While many of these features are readily available in my data, some others are hard to obtain considering the early time prediction demands for my task. Moreover, the prediction uses training set lasting a long period of time (several months) on a coarse time granularity—weekly volume, which clearly is not suitable in my study.

Another well-known research is done by Ma (Ma, 2013). He performed a thorough feature exploration for newly emerging hashtags popularity prediction. He modified the problem to a multi-class classification task to avoid direct prediction and make a comprehensive comparison among a wide range of feature sets for their predicting effectiveness. He did doing the prediction on daily basis, which is a finer time granularity and similar to my task. He also creatively adopted some context features derived from users mention network to improve the prediction, while in my work, I use similar ones from the retweet network. Thus his work is much more relevant to my purpose, but still does not precisely fit my situation. Firstly, the time frames and scales of the data are different. The data the author used are restricted to Singapore based users and have a time span of several months; while the data I have access to are generally world-wide tweets about specific urgent events and usually only proper about 1-2 weeks. With the long-term and relative homogenous (same set of users) data in hand, instead of a strict time based prediction—which is necessary for my implementation, the author simply performs a 10 fold cross-validation (CV) over hashtags occurring at different time stamps. This research explores the predictive power of wide range types of features on newly emerging hashtags prediction; but it is not in good alignment with what I aim to achieve: a time-based early-stage prediction with only limited information available within a short period of time. Also the author defined the number of users adopting a given hashtag as the popularity measure, but I decide to use the number of retweets a hashtag obtained as the target response to characterize the hashtag ability of spreading information under the urgent information diffusion context. This was also due to the differences between the goals of the prediction task. Moreover, granted the distinctions on data and prediction task, many features adopted by the author became either unnecessary or unavailable—e.g. the border user definition and the corresponding exposed vectors were less relevant given retweets as the target response. There were also other features not available for other reasons—using information from the full data set (the clarify scores) or requiring extra effort (hashtag lexical features).

As to more general and abstract topics, while considerable literature focuses on various Twitter topic modeling or detection techniques (Cataldi, 2010) (Hong L. a., 2010) (Mathioudakis, 2010) (Ramage, 2010) (Godin, 2013), fewer studies are seen on predicting the continuous popularity of these topics. Potential reasons might be the prediction task is not as intriguing as the topic modeling or other popular prediction problems—prediction of tweets popularity for example. For the actual topic definition, one state-of-art topic modeling to apply is Latent Dirichlet Allocation (LDA) (Blei, 2003). Many topic modeling studies simply adopt it or other variants of LDA models, including the ones mentioned above. However, LDA may not be the most suitable topic modeling method in my case. Yang (Yang, 2014) has a very good discussion about large-scale and high precision topic modeling in Twitter. Finding the best topic models is not the focus of my study. I use the LDA model simply as an extension study of the hashtag prediction in order to see how the prediction will vary between straightforward hashtag defined topics and more abstract model defined topics. Thus, in this study, I just use LDA to model the latent or general topics and then attempt to predict the popularity of the generated topics as weighted aggregation of documents (to be defined) popularity.

In short, although my proposed prediction tasks, the focus of my discussion in this study, share things in common with previous studies, these tasks still exhibit unique properties and thus require novel solutions.

1.3 Data sets

There are two types of data sets involved in my study: 15k_user data sets and event data sets.

1.3.1 15k_user data set

All of the data are collected through Twitter API (for more details see https://dev.twitter.com/overview/api), where users can specify various conditions (user status, key words, timestamps, locations etc.) to obtain the corresponding tweets information. The analyses are performed either with offline or online—streaming mode with different API options. Once the conditions for the queries are settled, the API server will responds with the query request—for offline mode it will return the batch data, for online mode it continuously will update with streaming data.

The data obtained contain all basic information for each tweet such as: status_id (unique identifier for each tweet), timestamp, text content, user_id (unique identifier for each user), user_name, basic social profiles of the user, links and hashtags information of the tweet etc. For my study, I have collected two types of data sets—the 15k_user data and events data sets.

The 15k_user data sets collect all tweets, followers and friends from a selected group of 15k users (Swaroop, 2014) with a period lasting about 2-3 months. The main motivation for the collection of 15k_user data sets is to find representative subsets of the whole Twitter universe—including all users, tweets and the associated network structure. Although collection of Twitter universe over a fixed period of time is not infeasible, but it does require additional cost from the Twitter API and often is not necessary for many applications. According to Swaroop (Swaroop, 2014), the following protocol is used to select the 15k users:

1. Find the first active user and add it to the active_user_list
2. Obtain followers of the active users, add them into the user_list
3. Iterate user from the user_list and determine if it is an active user, if yes repeat step 1 and 2.
4. Repeat the above user picking steps until the size of the active_user_list reaches 15k.

In the above steps, an active user is defined to satisfy: i) Having at least one retweet within his or her last 100 tweets; and ii) Having the tweet frequency not less than 1 tweet per day within his or her last 100 tweets. After the 15k active users have been selected, all of their activity (tweets), followers and friends are also obtained within a period of 2-3 months.

Two sets of 15k_user data are obtained for my analysis—15k_2011 and 15k_2012, which are collected from different months in 2011 and 2012 respectively, but the users are shared between the two sets. Since the collection only specifies users and does not put restriction on the tweets content, the data include everything the users talked about—news, real time events, normal chats, or any other conversations. This is in contrast to the event data set, which only includes tweets about a specific event.

The 15k_user data is mainly used for the study in Chapter 2, where I describe the generation of a subset of 15k_2011 set—the OBL data set. This data set is about the real big event—death of Osama Bin Laden and is used as the sample set for the diffusion tracking study. The OBL set is extracted from the 15k_2011 set using regular expression with some key terms derived from his name—Osama, bin laden etc. More details of the data are described in later sections.

1.3.2 Event data set

The event data sets have similar structure and data fields as the 15k_user data; but now the tweets are confined by selected key terms instead of selected users. These event data sets are used for the prediction study described in Chapter 3 and Chapter 4. With certain conditions specified—keys words, location or other restrictions by the end users, the Twitter streaming API is expected to return a collection of tweets fitting into those conditions. But according to the Twitter API document, the volume of the returning collection cannot exceed the volume of a small fraction of total public tweets stream at the same time. This is saying that if the volume of the target tweets collection is larger than the volume of a certain fraction of the total public tweets stream at the same time, only a sample of the target collection will be returned; otherwise, the full collection should be returned. Even under the condition that only a sample of target collection is returned, the sample itself usually already constitutes a huge amount of data considering the full volume of the public tweets stream, thus the effect of sample bias introduced in this way is limited. But Twitter also provides service to remove this volume cap with additional financial cost if that is desired.

Once an urgent event is identified, the streaming collection can be started using key terms specifying that event. Following this, 4 event data sets are collected, each of which is concerning a crisis event: Hurricane Irene in 2011, Hurricane Sandy in 2012, Nepal earthquake in 2015 and Jonas blizzard in 2016. Simple key words are used to track the tweets—for example the following terms are used to grab the Irene data set: #irene, irene, #hurricane, hurricane, #hurricaneirene, where the prefix "#" denotes hashtags in Twitter. This simple approach will inevitably catch irrelevant tweets and does not include all tweets talking about the events, but I believe the level of noise is low and the volume of the collection is large enough so that these bias factors do not affect the final results. I will come back to discuss more details on the data sets in later sections.

1.4 Network analysis

1.4.1 Retweet network

As mentioned in the previous section, retweets can provide valuable insights for information diffusion study on Twitter. Thus the network formed by retweets can be considered as the approximate to the underlying diffusion pathways. Moreover, the time stamps information associated with the retweets can be used to investigate the temporal properties of the diffusion. In contrast to the other often studied network—the F-F network, the retweet network not only better represent the diffusion pathways, but can also provide rich dynamic information about them. The definition of the retweet network is straightforward: if user A retweet user B, then I consider there is a directed edge formed between user A and user B (with the timestamp of the retweet as the edge attribute), thus a directed network is formed.

Table 1.2 shows one example of retweet with several common data fields. Generally a retweet can be captured by either the retweet indicator or using the text regular expression "RT @username". However as seen in this example the retweet indicator does not work and cases like this are not in a small number, thus I decide to use the regular expression to capture all the retweets. In the above example I will form a directed edge from user GLB62 to user keithurbahn.

From now on, network will refer to the retweet network by default unless otherwise specified.

Date	**Status_ID**	**User_ID**	**User_name**	**Timestamp**	**Text**	**Retweet indicator**
5/1/2011 22:27	64878655444234200	14944471	GLB62	1304303251	WOW RT@keithurbahn: So I'm told by a reputable person they have killed Osama Bin Laden. Hot damn.	0

Table 1.2 One example of retweet from OBL set

1.4.2 Related network statistics

Given the retweet network, I have chosen several important network statistics for the study. Since the network is directed, I choose network statistics that are properly defined for directed network.

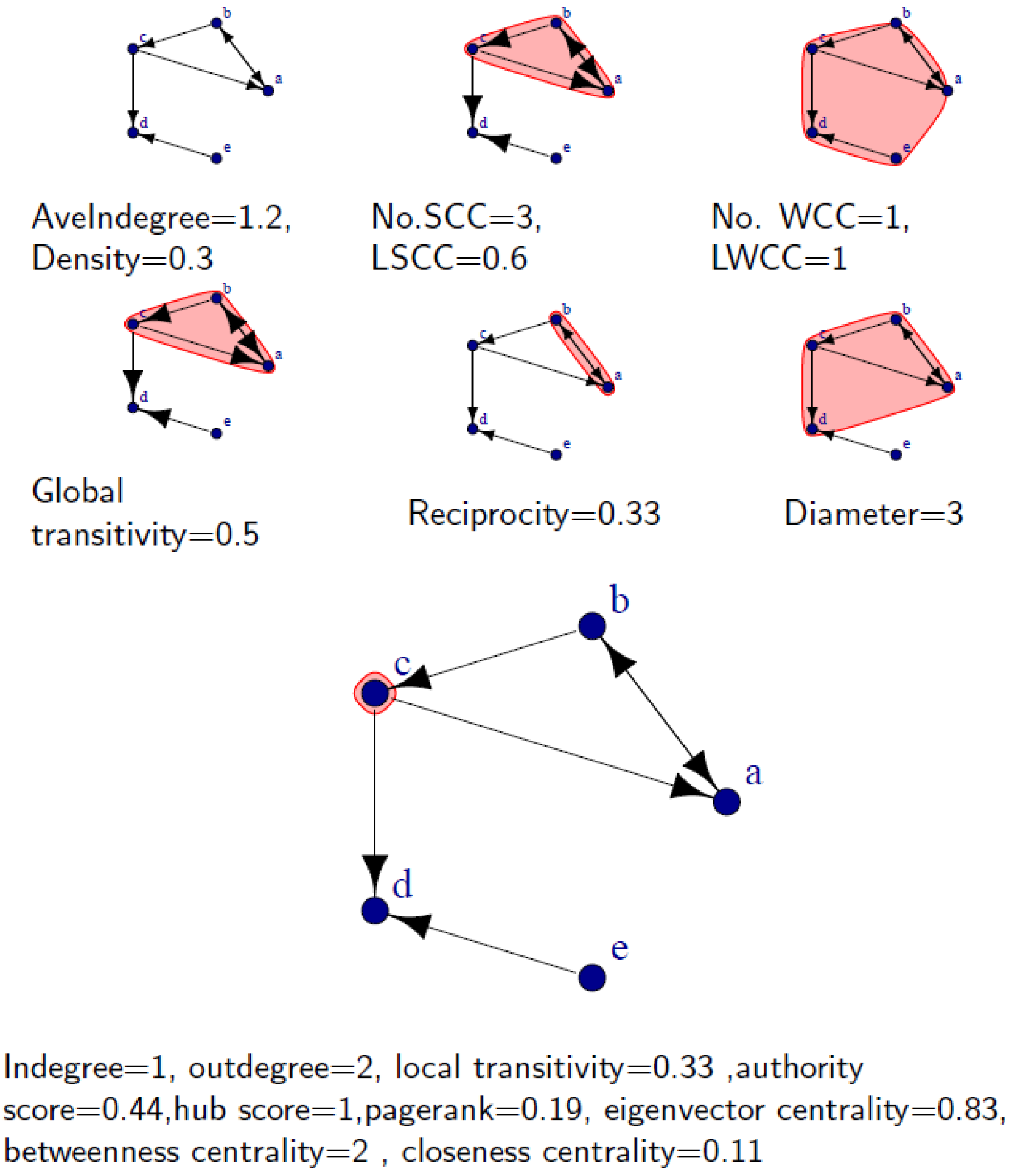

b
c
a
d
e
AveIndegree=1.2,
Density=0.3
No.SCC=3,
LSCC=0.6
No. WCC=1,
LWCC=1
Global
transitivity=0.5
Reciprocity=0.33
Diameter=3
Indegree=1, outdegree=2, local transitivity=0.33 ,authority score=0.44,hub score=1,pagerank=0.19, eigenvector centrality=0.83, betweenness centrality=2 , closeness centrality=0.11

Figure 1.2 Global level and node level network statistics from a simple network

Figure 1.2 gives a picture illustration of the network statistics involved in my study. A simple directed network with 5 nodes and 6 edges (each arrow in the graph is a directed edge) is used to explain various network statistics. In summary the network statistics studied can be classified into two categories: global level statistics and node level statistics. Global level statistics refer to statistics defined on the global or the whole network level and node level statistics refer to statistics defined on the node or user level.

More specifically, global level statistics include the followings:
1. Average degree: since an edge is directed, each link can be considered as an inbound link or an outbound link depending on which node is considered as the reference; but the average degree over all the nodes will be the same no matter inbound links or outbound links.

2. Density: it is a measure for how complete a network is connected—the number fraction of connected edges over the maximum possible edges.

3. SCC (WCC), LSCC (LWCC) and reciprocity: SCC and WCC are strongly connected components and weakly connected components respectively. For directed network a strong link exists between two users if two users can be connected in both directions, while it is a weak link if the connection only exists in one direction. All nodes that are strongly (weakly) connected from each other will form a strongly (weakly) connected component. LSCC (LWCC) is largest strongly (weakly) connected component, which is the fraction of size of largest strongly (weakly) connected component over the total size (number of nodes). Reciprocity is simply the counting fraction of edges that are reversible (bi-directional).

4. Global transitivity (clustering coefficient) and diameter: global transitivity counts the fraction of connected triangles over all connected triples, here I only consider undirected connections for simplicity and isolated nodes are not counted. The diameter is the longest path length (length of shortest path) between two nodes existing in the graph, also only count undirected connections for simplicity.

For node level statistics:
1. Indegree and ourdegree: simply counts of inbound links and outbound links for each node.

2. Authority and hub scores, eigenvector and pagerank: these are all centrality measures calculated based on adjacency matrix of the graph, which can be considered as "deeper and extended" measures instead of one-step measures of indegree and outdegree.
i) Authority and hub scores are computed recursively based on each other, which characterize the value of a node by inbound links and outbound links respectively (Kleinberg, 1999).
ii) Eigenvector centrality is derived from the principle eigenvector of adjacency matrix, and again the direction of edges is ignored in the calculation due to sparsity problems of the graphs.
iii) Pagerank is a popular centrality measure to quantify importance of web pages, what makes it different from the rest three is that it has an external parameters to be

determined—the damping factor. This factor is used to characterize the decaying influence over long-ranged links, which is usually set to be 0.85 for convention.

3. Local transitivity (clustering coefficient): similar to the global transitivity, it counts the fraction of triangles over connected triples from a given node. Isolated nodes are set to be zeroes and direction is also not considered.

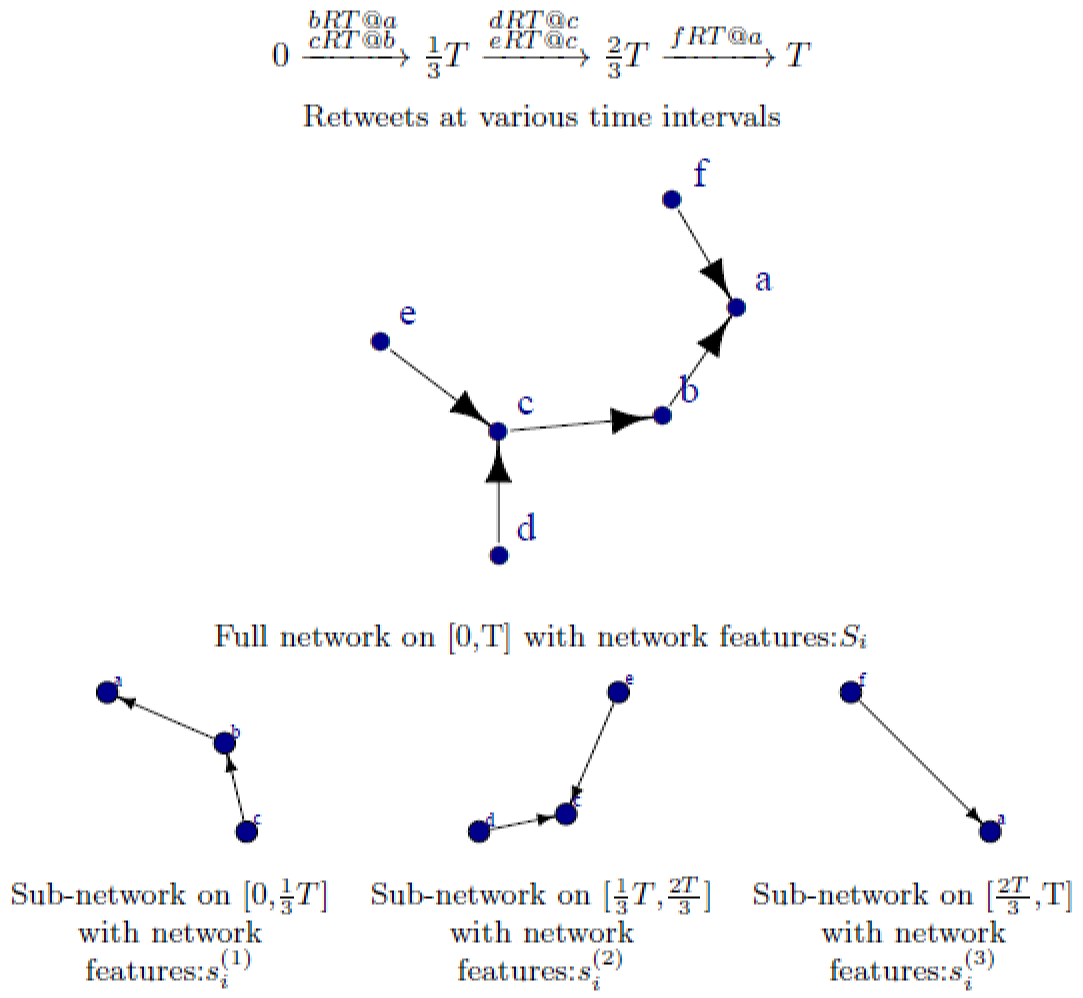
4. Betweenness and closeness centrality: these two measures are related to shortest paths among nodes. Closeness centrality is proportional to inverse of the average path length (length of shortest path) between the given node and the rest ones. Betweenness centrality simply counts the number of shortest paths through a given node.

All the above statistics are computed using the R package—igraph. Firstly the graph is constructed from an R data frame formed by the edge-list. Then different network statistics are computed using different built-in functions included in the igraph packages, further implementation details are illustrated in the manual of the package.

1.4.3 Dynamic network statistics extraction

In order to study the dynamic properties of the diffusion process, instead of using just the aggregate (static) version of all these statistics, I also need to take statistics from snapshots of temporal networks based on the timestamps. The idea is straightforward, and Figure 1.3 demonstrates the statistics extraction steps from a simple network.

In Figure 1.3, I assume to have the network below within time period $[0, T]$, and the edges are formed within each sub-period: e.g. user d retweet user c and user e retweet user c within sub-period $[\frac{1}{3}T, \frac{2}{3}T]$. Then within each sub-period, a sub-network is formed and the corresponding statistics are extracted, I will get one set of statistics within each given time interval and in the sections of this dissertation, I just refer these statistics as dynamic network statistics, while the statistics derived from the whole period $[0, T]$ (the full network) will be called static statistics—implying they do not contain the temporal information of the network.



$$0 \xrightarrow{\substack{bRT@a \\ cRT@b}} \frac{1}{3}T \xrightarrow{\substack{dRT@c \\ eRT@c}} \frac{2}{3}T \xrightarrow{fRT@a} T$$

Retweets at various time intervals

Full network on [0,T] with network features:S_i

Sub-network on $[0,\frac{1}{3}T]$ with network features:$s_i^{(1)}$

Sub-network on $[\frac{1}{3}T,\frac{2T}{3}]$ with network features:$s_i^{(2)}$

Sub-network on $[\frac{2T}{3}$,T] with network features:$s_i^{(3)}$

Figure 1.3 Dynamic network features extraction illustration

The network truncation time T defines the full network I would like to study. With the additional parameter—the time window size (denoted as Δ), which is $\frac{1}{3}T$ in my example, I can control how transient or persistent temporal information I want to know from those dynamic statistics.

The example in Figure 1.3 just shows one way of defining the dynamic statistics, which I call it segmented dynamic statistics. The other way to extracting the dynamic statistics is to define the cumulative networks over time (Figure 1.4) and I can obtain the cumulative dynamic statistics.

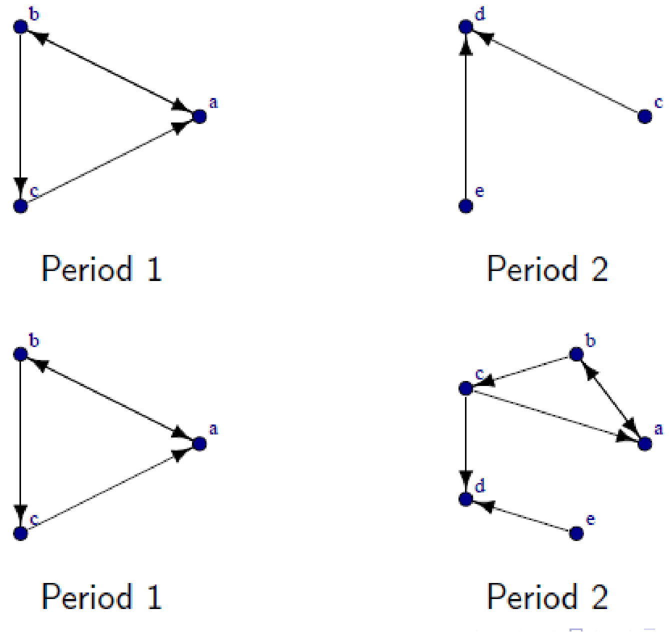


Figure 1.4 Two types of dynamic statistics series—segmented series (top) and cumulative series (down)

The potential problem with the cumulative series is that the statistics series are correlated (future values depend on past values) and it is not good for predictive analysis—the correlated features will cause problems especially for linear models. Thus I will stick to the segmented series from now on unless otherwise specified. But there are still situations when I need to use the cumulative series: one is when the segmented sub-graphs are very sparse and cause the segmented statistics less informative or even not valid; the other may just be I do want to incorporate the cumulative effects into my observation.

1.5 Outline of work

In this dissertation I want to demonstrate how static and dynamic network properties of the retweet network reveal valuable insights of urgent event diffusion on Twitter, the outline of the work is listed as follows:

In Chapter 2 I focus on how I use these network statistics to perform real time tracking of urgent event diffusion and how they can shed lights on various temporal patterns of the diffusion process. Moreover, I make a simple attempt to compare whether these statistics can differ among different types of popular topics on Twitter—trends (Mathioudakis, 2010), which may not be real life events. All the analysis done in this chapter used the 15k_user data since only these data include trends of various types and is not restricted to real life urgent events.

In Chapter 3, I describe how I aim to spend my effort on one important problem in the field of information diffusion study—the user influence prediction. I take the size of retweet cascades as a measure of influence and established an early time prediction framework using a "cross prediction" mode to make effective prediction on the user influence within only limited amount of prior information given. I also study how various experiment factors affect the prediction performance and I reach relatively consistent results. In this study, I use the event data sets since 15k_user data only include a small subset of the whole diffusion of the event.

In Chapter 4, I move to the prediction of popular topic during urgent event diffusion. The same data sets are used and similar prediction framework as the user case was applied. I adopt two types of topics to perform my prediction tasks: self-defined topics, i.e.,—hashtags, model defined topics, i.e., —latent topics. I observed similar results with the topic prediction as to the user case.

Chapter 5 is a comprehensive conclusion of the whole study and gives promising pathways for future extension.

Chapter 2: Tracking urgent event diffusion on Twitter

2.1 Introduction

2.1.1 Motivation

The retweet network is suitable to explore the temporal patterns of information diffusion on Twitter. Given all the information provided by the Twitter API, the retweet network can serve as a real time monitor of information diffusion on Twitter—not just for diffusion of events, but for diffusion of any popular topics—which are called trends on Twitter. By viewing the event diffusion process as expansion of the retweet network, the diffusion can be tracked and analyzed with retweet network analysis. In this Chapter, I show that the dynamic network statistics derived from the retweet network can reveal detailed dynamics of urgent event diffusion. Moreover, these statistics can also be used to differentiate hashtags of different types—e.g. hashtags related to real life events or hashtags related more general topics on Twitter.

2.1.2 Data description

For the diffusion tracking study, I mainly use the OBL data set from the 15k_user data as my sample data set. Similar results are derived from the event sets.

Data set	**No.tweets**	**No. users**	**No.retweets**	**No. mentions**	**No.tweets with URLs**	**No.tweets with Hashtags**
15k_2011	10979280	14623	2426257(0.221)	6879875(0.627)	3510811(0.320)	2154061(0.196)
15k_2012	7850583	12043	1806645(0.230)	4577257(0.583)	3487325(0.444)	2327301(0.296)

Table 2.1 15-k user data summary

Table 2.1 gives a summary for the two 15k_user data sets I have analyzed, the time frames are from 25 Apr 2011 13:24:57 GMT to 17 Aug 2011 01:23:22 GMT for the 2011 set and from 07 Sep 2012 17:00:04 GMT to 11 Dec 2012 11:56:23 GMT for the 2012 set. The numbers in the parentheses represent the corresponding fractions over the total number of tweets. Not all users are active within the periods. While fractions of retweets and mentions remain stable, the fractions of URLs and hashtags have increased—implying users are getting more used to links and hashtags.

Figure 2.1 shows the overlay comparison of F-F distribution between the two 15k_user sets, only users (11966) from the intersection set of 15k_2011 users and 15k_2012 users are shown in the figure. Most users do not change their behavior much, though some users shift a little bit. The number of tweets has generally increased. For both 2011 and 2012 sets there is an unusual turning slope, due to the Twitter restriction on the number of friends a given user can follow—the limit of the friends depends on the number of followers a user has. From the plot it seems that this is simply a linear relationship and it is possible to infer the approximate linear equation from the data. Also a user is likely to post more tweets with higher number of followers, that could be because that posting more tweets and having more followers could have a reinforcement effect on each other.

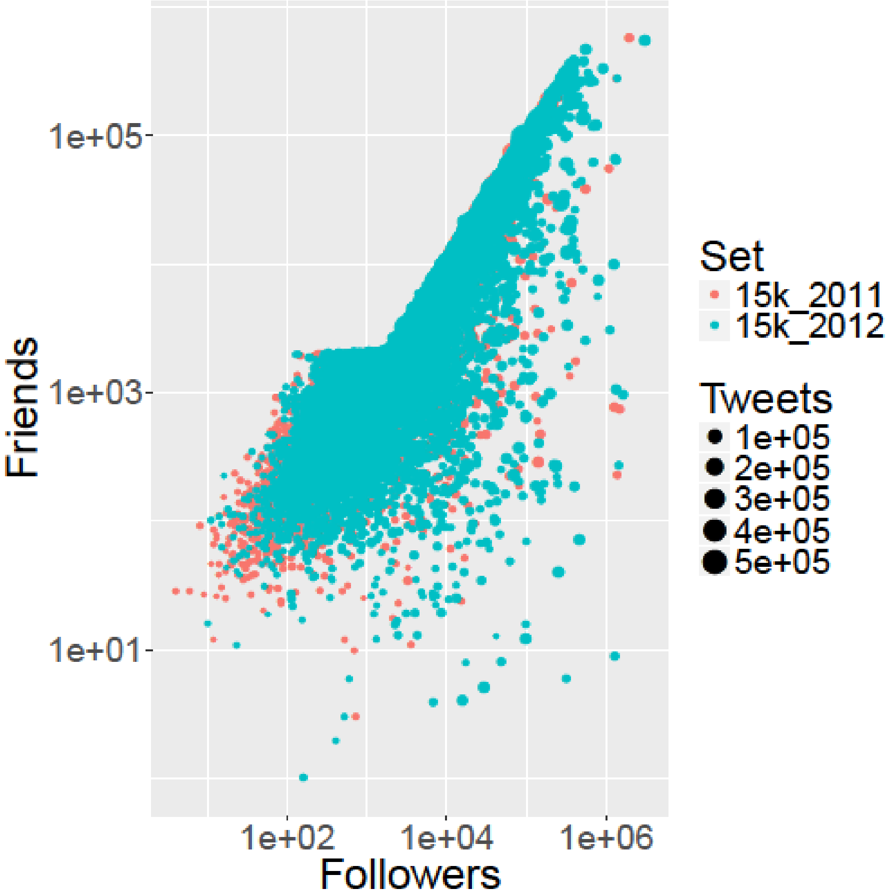


Figure 2.1 15k_user data sets F-F distribution comparison

Table 2.2 is a comparison table for the OBL data and the background set—which is simply a mixture of everything (all tweets) from the 15k_user data but with the same time frame as the OBL set: from 02 May 2011 02:27:10 GMT to 04 May 2011 02:23:51 GMT.

Data set	No.tweets	No.retweets	No. mentions	No.tweets with URLs	No.tweets with Hashtags
Background	470438	114206(0.243)	288058(0.612)	174262(0.370)	120052(0.255)
OBL	28743	12815(**0.446**)	16338(0.568)	14076(**0.490**)	7666(0.267)

Table 2.2 OBL data summary

The OBL set, which is the representative for the event data set, has significantly higher fractions of retweets and URLs compared to the background—implying more retweets and links occur during urgent event diffusion compared to the normal-level Twitter activity. While the components of hashtags and mentions do not differ much, showing these two types of tweets are not well distinguishable between urgent event diffusion and normal activity.

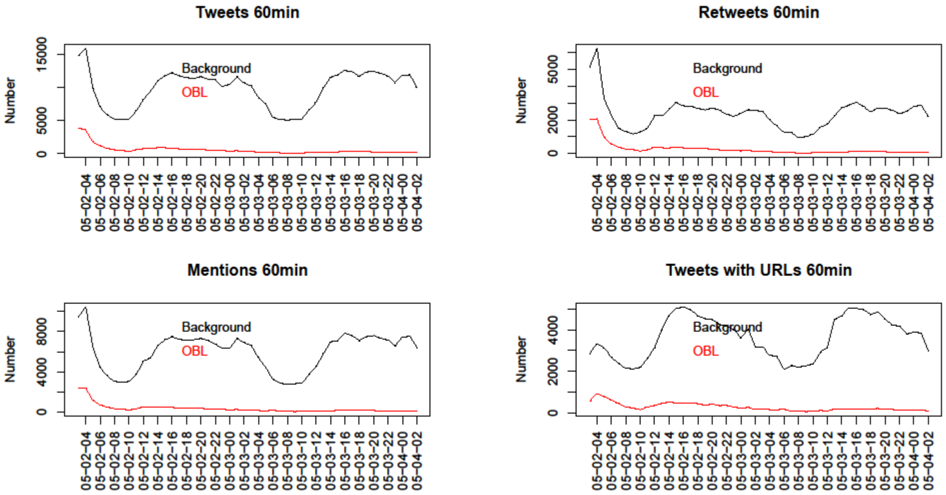


Figure 2.2 Evolution plot for OBL and background sets

Figure 2.2 is the evolution plot for OBL and background sets, and the time interval length for the series is 60 min. It clearly shows how the diffusion dynamics of an event differ from the normal-level activity. All curves from the background set have clear periodical patterns, while ones from the OBL set slowly decay. All types of tweets follow the similar trend for both sets, only differing by their scales.

Statistics	Background network	OBL network	OBL_15k network
No. nodes	61963	10099	754
No. edges	114206	12815	787
Avg. indegree	1.843	1.269	1.044
No. SCC	61397	10088	743
No. WCC	2118	1110	146
LSCC	0.00749	**0.000495**	0.00663

LWCC	0.873	0.711	0.561
Transitivity	0.00513	0.00223	**0.0193**
Reciprocity	0.00687	**0.00141**	**0.0234**
Diameter	29	8	7

Table 2.3 OBL retweet network summary

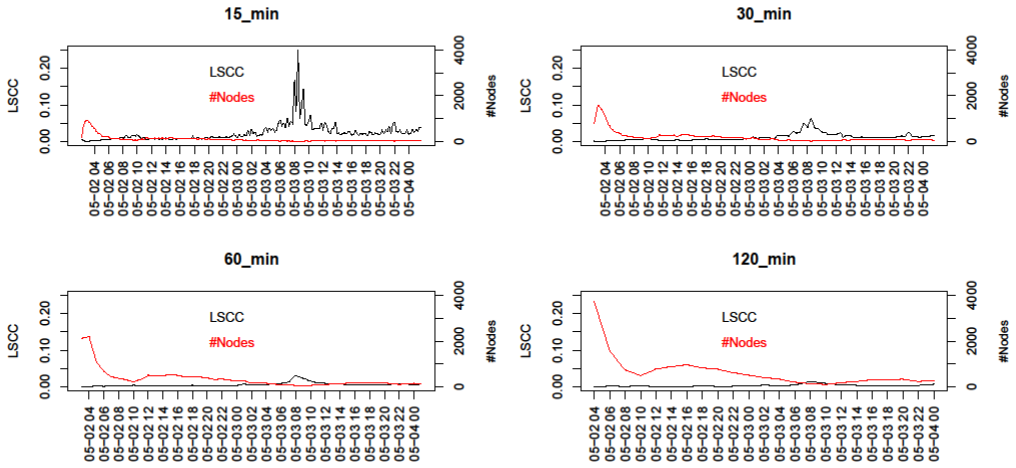
Table 2.3 shows the retweet networks comparison, where the OBL_15k network refers to the network only including edges between the 15k users. The three networks are ordered in decreasing scopes, and also form a decreasing subset series—the latter ones are subsets of preceding ones. Besides the differences on the scales of the network, the OBL_15k network is better connected as can be shown by the transitivity and reciprocity. But the networks are not generally well connected in the strong sense and do not have much clustering patterns (number of triangular links). By looking at the retweet networks formed by other subsets similar results are observed; thus this can be considered as general properties of the retweet network: (1) few mutual connections (users have been retweeted by each other) and clusters, and (2) mainly composed of unidirectional links. All these values are even lower in the event network—the OBL network, which means this property is reinforced during event diffusion.

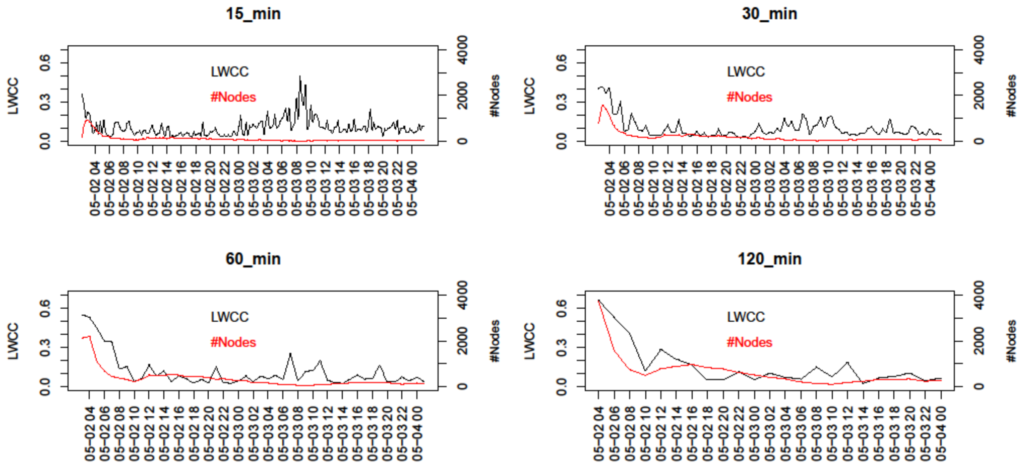
2.2 Event evolution tracking

2.2.1 Global level tracking

In the next two sections I will describe the various network statistics that are used to track the diffusion of the event over time, the default network used will be the OBL network unless otherwise specified.

Figure 2.3 shows the evolution plots of various global statistics of the OBL network, the titles of the subplots are the corresponding time window sizes. Both the number of nodes and number of edges are decaying gradually--implying the diffusion is gradually ceasing. The curves become smoother with a larger time window; local patterns average out as some small peaks in the plots have disappeared with increasing time window size. The peaks at 8AM of May 3rd in the LSCC, LWCC and reciprocity plots demonstrate there are some mutually connected links at that time, but the number of the links is not large given the values of the statistics. Except for some small peaks, the reciprocity remains zero, showing barely any edges connect both directions. In the big picture, it shows the network is weakly connected relatively well but not strongly connected. This can also be seen from the differences of the trends between the LSCC and LWCC curve: while the LSCC curve remains nearly flat except few bidirectional edges, the LWCC curve exhibits similar patterns as the total number of nodes in the long run. This indicates the largest weakly connected component follows nearly the same pace with the number of nodes, which can also be viewed as the sum of size of all components. To put this in another way, it shows that the diffusion starts out with large-size weakly connected components—many users are connected by the retweet paths, then gradually dies out with fragments of small size weakly connected components—much fewer users continue their discussion in smaller groups; only few users have retweeted each other in the whole procedure—meaning the flow of the information mainly goes in one direction. In summary, by controlling the time window size and the statistics types, I can reveal both local and global dynamic patterns of the diffusion to a specified level of resolution.





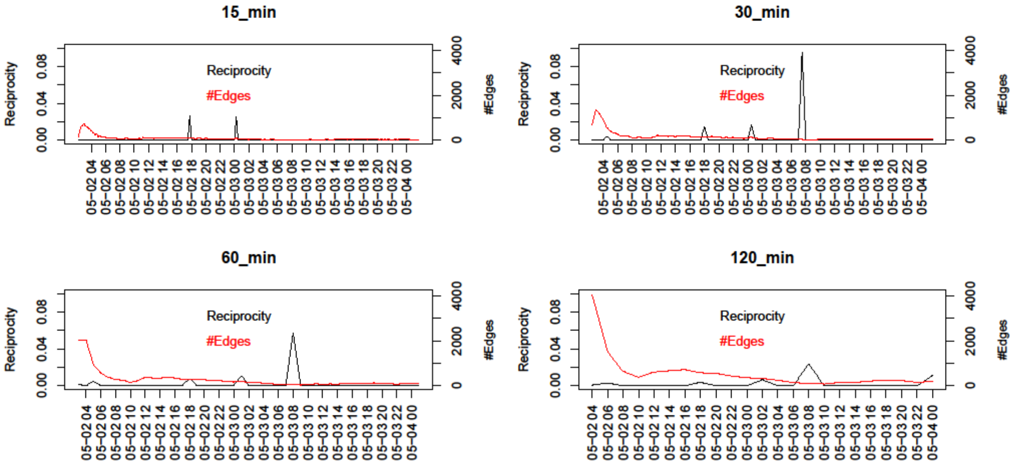


Figure 2.3 Global level network statistics evolution plots for OBL network

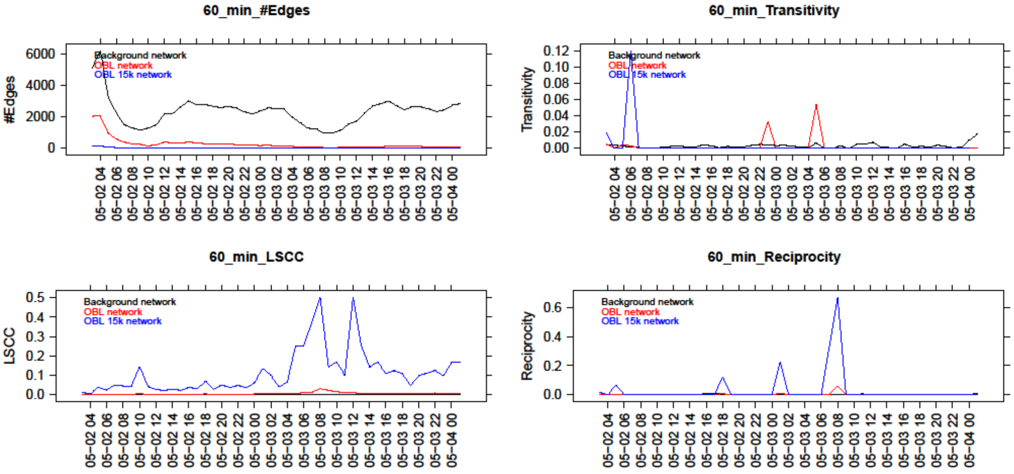


Figure 2.4 Global level network statistics evolution plots comparison for three networks

Figure 2.3 has illustrated the comparison varying the statistics and the time window size, Figure 2.4 will demonstrate the comparison from different scopes of networks. Apart from the obvious differences on the scale, clearly both the strongly connected and clustering patterns stand out when the scope of the networks is decreasing. This is a natural result from the fact that only retweets originated from the 15k users are included in the data—meaning the mutually connected links can only come from the 15k users. This result also discloses a flaw in the 15k_user data—it naturally carries a bias when studying the retweet network by excluding all retweets not coming from the 15k users. Thus this diffusion study will inevitably be biased, but similar results are likely to manifest by observing the patterns within the 15k network, which is free of bias. Moreover, even if the results are incorrectly specified due to the underlying bias, the methodology still proves its value and that is the main idea I want to convey in this section.

2.2.2 Node level tracking

The previous section mainly discuss how I use various global level network statistics to track detailed dynamic patterns of diffusion, in this section I will extend similar discussion to node level statistics.

While the global level statistics can tell explicit information about the global patterns, it is natural to believe the node level statistics can also be informative for node level tracking.

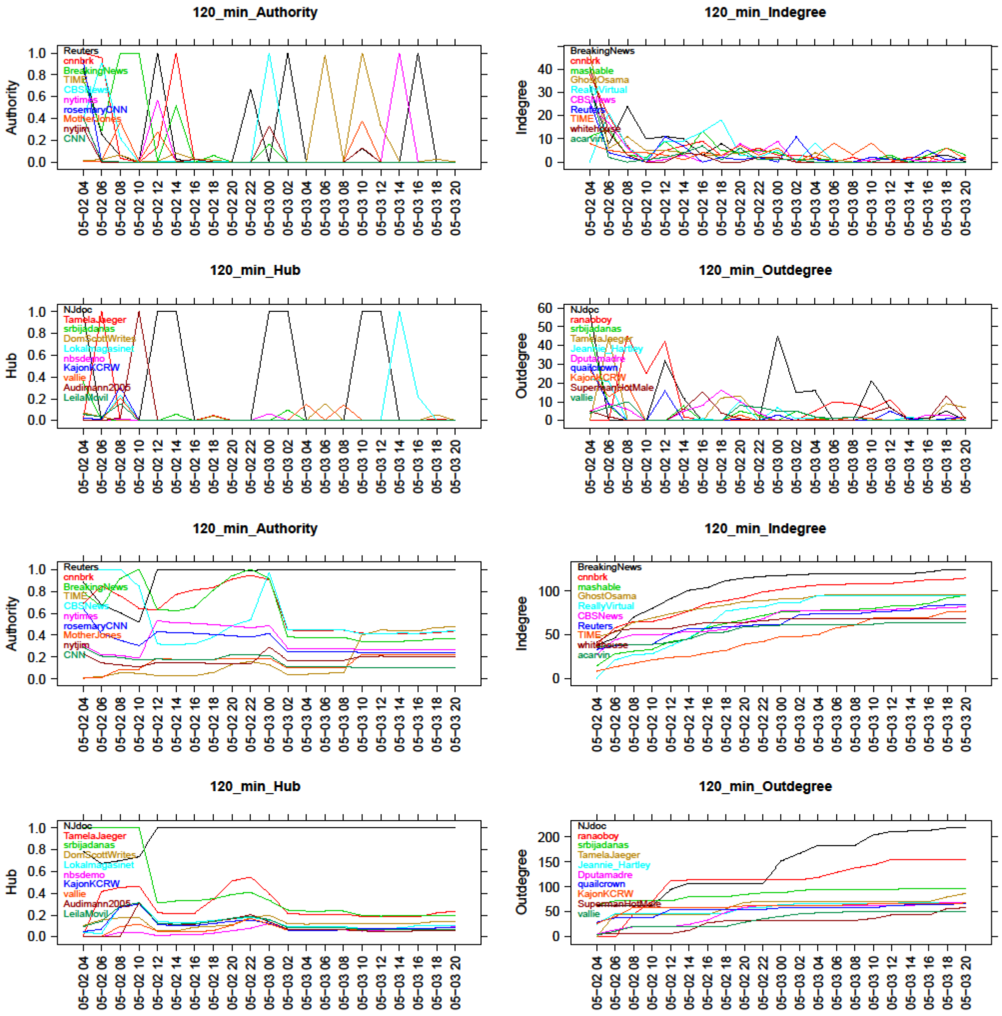


Figure 2.5 Node level network statistics evolution plots for OBL network

Figure 2.5 shows the top 10 node level statistics evolution plots ranked by 4 different centrality measures, the upper 4 plots use the segmented series and the lower 4 plots use the cumulative series for comparison (for the differences between the two series, please check the dynamic network statistics section). The top 10 nodes are different with different centrality measures, although authority score and indegree, hub score and outdegree are closely related from each other. The most valuable information obtained from the figure is that the dynamics of important nodes can be monitored, where the importance is defined by various centrality scores. The segmented series plot emphasizes the nodes importance within each selected time window, while in the cumulative plot I can see how the cumulative importance ranking is changing over time. Considering the confusion from plotting too many lines, only 10 nodes are shown for the node level plot, and these are the top nodes that are needed to be studied. But this methodology provides the flexibility where nodes can be targeted and monitored as decided. All nodes can be monitored at the same time, or target top nodes ranked by a specific centrality measure.

2.3 Event hashtag identification

2.3.1 Hashtag evolution comparison

I have shown that these different network statistics can be used to shed light on the dynamics of urgent event diffusion. Now I would like to further investigate whether a real time event will exhibit distinct network properties to general popular topics on Twitter—trends or not. I think of this as the extension of the diffusion tracking study not only since they are closely related, but also because I have the full activity of the 15k users on general trends (not just ugent events).

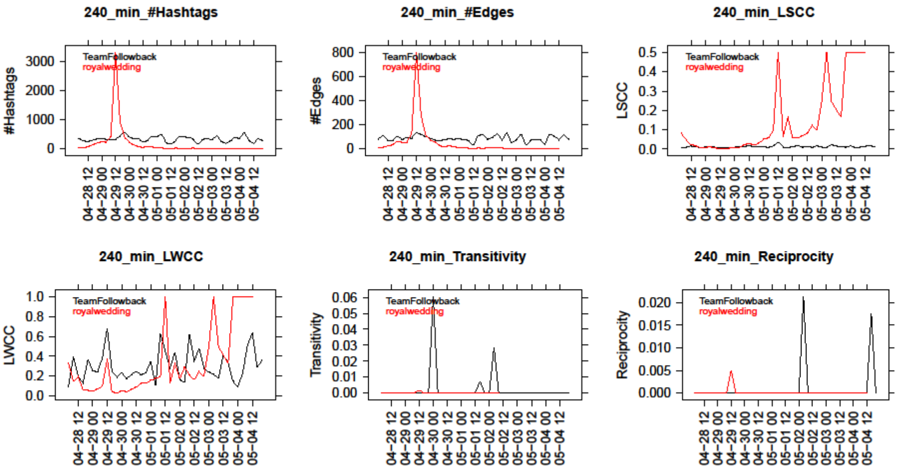


Figure 2.6 Evolution plots for event and non-event hashtag networks

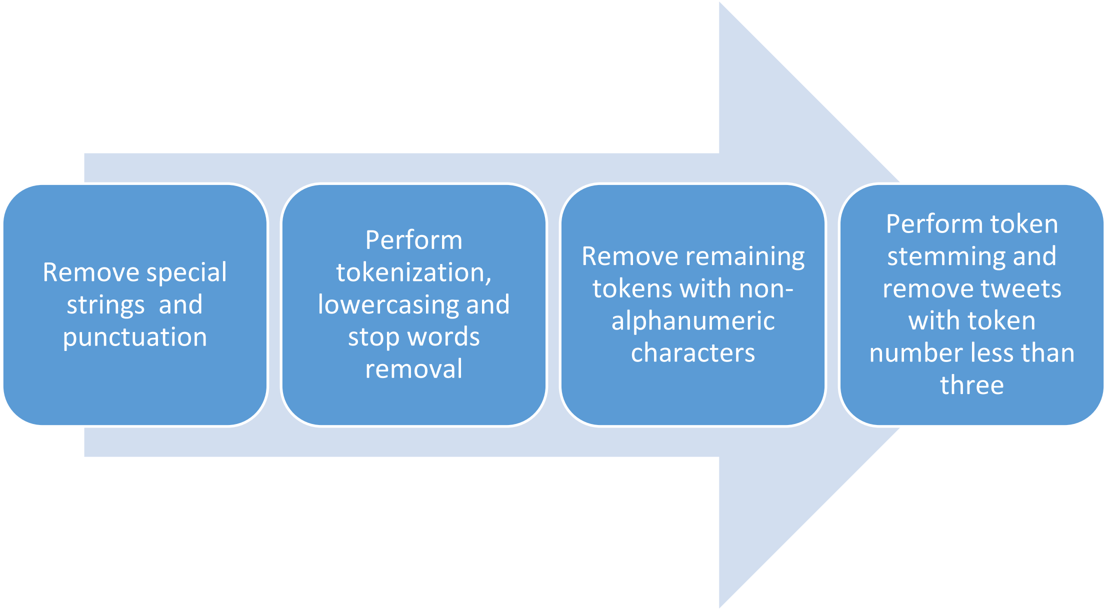
In Figure 2.6 I select two hashtags of different types and then compare the retweet network properties derived from the two hashtags. "TeamFollowBack" is a popular meme on Twitter, using this meme, interested users will adopt the hashtags, follow each other and communicate each other. Hashtag "royalwedding" corresponds to the British royal wedding in 2011. The first hashtag can stand for a large group of trends on Twitter which are not related to life outside Twitter and I call this a "meme hashtag", while the second hashtag serves as an indication of an event happening outside Twitter and I call this an "event hashtag". Clearly both the hashtag volume (number of tweets with a given hashtag) and the number of edges (the hashtag retweets volume) show the differences between these two types of hashtags—the meme hashtag does not have a clear trend and remains relatively stable over time while the event hashtag follows a sharp burst-decay trend. Other plots also show the differences: the LWCC and LSCC of the event hashtag network have much larger variations than the ones from the other network. But the meme hashtag network has more clustering and mutual links since it includes more mutual communication; this is in contrast to the event hashtag network, which mainly contains simple one-directional links. Similar results also happened in other event related hashtags.

The above comparison raises the conjecture that hashtags related to real life events might be distinguished from other types of hashtags using the network statistics. Although I do not have many event related hashtags in the whole 15k_user data, this is a potential future research direction.

2.3.2 Hashtag clustering

Following the conjecture from the previous section, I make an attempt to cluster a group of selected hashtags using network statistics. The clustering results are compared to the results obtained using the hashtag tweets text, which is considered as the ground truth here. The selected hashtags are from both 15k_user data sets:
15k_2011 set:
'royalwedding','Osama','OBL','BinLaden','Tornado','Joplin','MemorialDay','TeamFollowback','FF','NowPlaying','YHP','SocialMedia','MADNESS','quote','SHOUTOUT','500aday','fb‘

15k_2012 set:
'election','election2012','iphone5','debate2012','blackfriday','hurricanesandy','halloween','hurricane','facebook','dope','twitter','business','hiphop','travel','jobs','music','marketing','tech','hr'



Remove special strings and punctuation

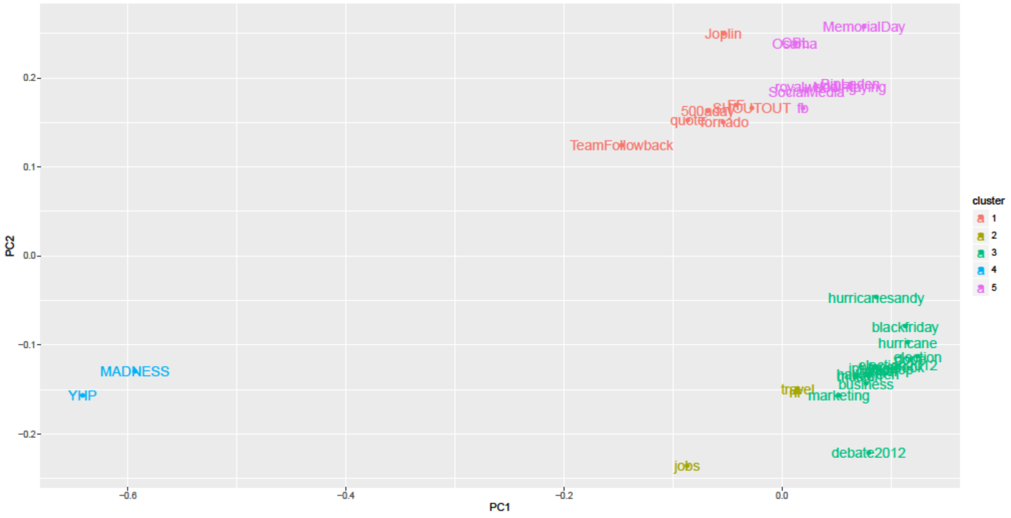
Perform tokenization, lowercasing and stop words removal

Remove remaining tokens with non-alphanumeric characters

Perform token stemming and remove tweets with token number less than three

Figure 2.7 Hashtag tweets text cleaning and processing

Here I intentionally choose a balanced mixture of event related hashtags (boldfaced) vs. non-event ones to see how they are different from each other. Here I have tried my best to find hashtags related to real events. The same list of hashtags will also be used in the binary classification task for the next section. The network features used for both tasks are: average indegree', 'density', 'LSCC', 'LWCC', 'WCC ratio' and 'transitivity'. The WCC ratio is the ratio of number of weakly connected components over total number of nodes. To accommodate the variations among time spans of different hashtags, I choose the uniform time span of 6 days for all hashtags, where starting times are the times when the hashtags are first adopted. Also the time window size is chosen to be 1d to obtain various dynamic network statistics, both the static statistics and the dynamic ones are combined to perform the clustering.



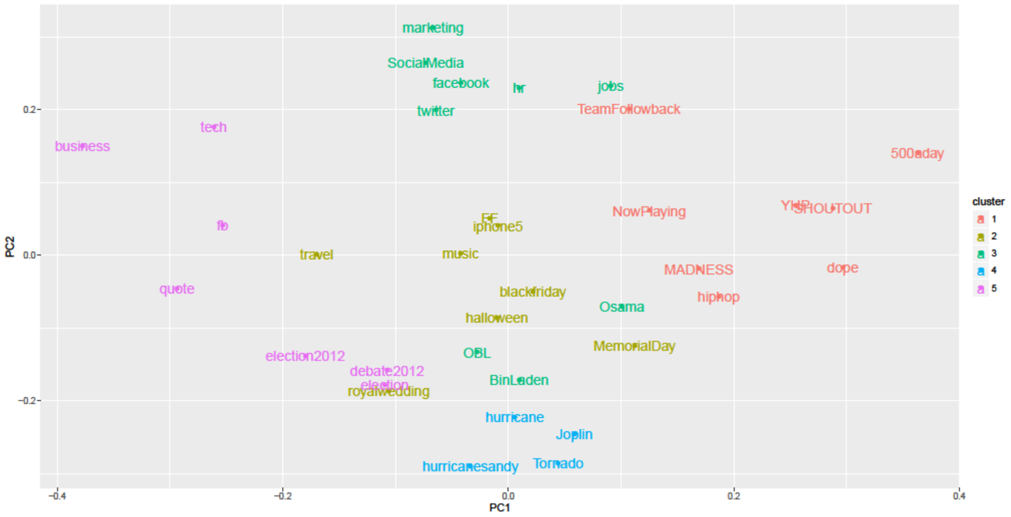


Figure 2.8 Hashtag K-means clustering 2D principle components representation (K=5, the upper one uses the network statistics, the lower one uses the text features)

I adopt the simple but powerful K-means clustering method (Hartigan, 1979), which is an iterative clustering method trying to minimize within-cluster distance given the clusters number.. Figure 2.7 shows the full text processing flowchart for hashtag tweets text, which are pretty standard text cleaning steps. Here the text for each hashtag refers to the text of all types of tweets (retweets or mentions etc.) from a given hashtag. The obtained tokens after the cleaning process are further filtered by selecting the top 10k tokens weighted by TF-IDF (term frequency inverse document frequency) (Blei, 2003) , which is a popular weighting strategy for document related analysis such as topic modeling. All the text cleaning steps are done using Python library NLTK. The special strings to be removed, i.e. mainly URLs, are defined using regular expressions. The NLTK ordinary word tokenizer was used and the stop words list from NLTK is modified a bit by adding letter "t" to all negation stop words, doing so is because I will perform punctuation removal before stop words removal. The stemmer used is the Snowball stemmer. The same process will be used for topic prediction task in Chapter 4.

Figure 2.8 demonstrates the clustering results (K=5) with a 2D principle components representation, which is using the first two components of principle component analysis (PCA) to visualize the clustering results. First I look at the results from text features (the lower plot), some closely related pairs are close from each other, which is really making sense: SocialMedia, twitter and facebook; election, election 2012 and debate2012; Osama, OBL and BinLaden; hurricane, hurricanesandy, Joplin and Tornado etc. While for clustering results using network statistics, they are more concentrated. This is mainly due to the difference between the dimensionality of the data space: the total number of network features is only 42, and the number of text features is 10k. In spite of the great dimensionality difference, some meaningful pairs can still be obtained: Osama, OBL and BinLaden; Joplin and Tornado; hurricane and hurricanesandy; election, election 2012 and debate2012. There are still some discrepancies, especially to those non-event hashtags, but it is clearly demonstrating that the network statistics do have the ability to distinguish hashtags to some extent.

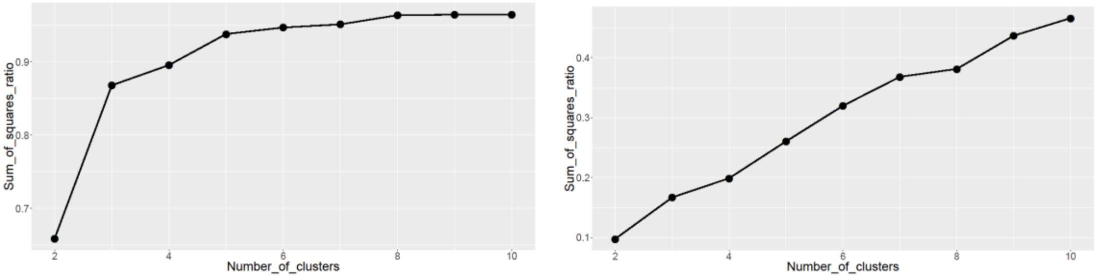


Figure 2.9 Hashtag K-means clustering sum of squares ratio varying by number of clusters (the left one uses network features, the right one uses text features)

Usually the performance of the clustering is measured by some external ground truth evidences, but since I do not have them here, I use the internal metric instead—the sum of squares ratio. The ratio is defined as the fraction of sum of squares among clusters over the total sum of squares, thus higher ratio will imply higher fraction of between-cluster variation—meaning better clustering performance. Figure 2.9 shows the sum of squares ratio changes over number of clusters from K=2 to K=10. While the ratio becomes saturated after K=5 for clusters from network features, it is continuously increasing for clusters from text features. Again this is mainly due to the high dimensionality of the text features, but it is also indicating that the hashtags are difficult to cluster generally--there has been much noise in the data.

Therefore I have observed that using network statistics can indeed provide some meaning results, however due to the noise of the overall data, it seems hard to obtain a good clustering even using the text features based on the internal metric—sum of squares ratio. In the future work, I will explore using subject matter expertise to provide an external ground truth. This will create the ability to validate whether using text features only is a good choice or not. In this case, I conjecture that the network features may provide additional improvement on the clustering and a combination of network features and text features could offer better results.

2.3.3 Event hashtag classification

To verify how network features can be used to distinguish event related hashtags versus non-event ones, I performed a simple binary classification task in addition to K-means clustering. The same set of hashtags and features are used in this task. The two classes are specified as event ones (boldfaced) and non-event ones. The AUC (area under curve) score is used as the performance metric since this is a normal balanced class classification, and AUC works well in that context. For a classification task with imbalanced classes, AUC is no longer a good score and other alternatives should be used (more discussion on this in later sections). The classification method used is deep learning (deep neural network), which is a state-of-art nonlinear method, the details of the method and corresponding parameters setting will be mentioned in later sections. The reason I am using this method here is due to the high dimensionality of text features, the ordinary linear model cannot handle this problem well due to singularity issues; another alternative state-of-art nonlinear method—random forest was examined, but the results are not presented due to computational costs.

Features	min	1st Qu	median	mean	3rd Qu	max
Network	0.35	0.75	0.85	0.83	0.91	1
Text	0.3	0.85	0.95	0.89	1	1
Combined	0.55	0.85	0.95	0.92	1	1

Table 2.4 Event hashtag classification average AUC scores with 100 repetitions

I propose exploring the task using the standard binary classification settings:
1. Split the data into 3: 1 ratio of training vs. testing sets with the same class labels ratio as the original data.

2. Train the model on the training set, then apply the trained model on the testing set, and measure the AUC score.

3. Repeat the random splitting in step 1 by n=100 times and all the steps above, collect all the AUC scores.

Table 2.4 shows the results from the classification. The performance of the classification highly depends on the training sets splitting, this is mainly due to the sparsity and labeling of the data—I need more well labeled data to better train my model. In spite of the large deviation in the results, there is still a slight improvement with the addition of network features, and using network features alone can already give comparable results. Considering the dimensionality difference between the two types of features, I believe the network features are indeed able to identify event hashtags well. However, due to the limited amount of data I have, I am not able to further validate this hypothesis.

2.4 Discussion

2.4.1 Summary

This Chapter mainly discusses how I utilize the 15k_user data to study the event diffusion on Twitter. In the event diffusion tracking section, I use various evolution plots to demonstrate how different network statistics can be used to reveal detailed diffusion patterns of an event. There are many parameters that can be finely tuned in order to deliver the most accurate information I want. The time window size is the one that controls the level of precision: a smaller time window size will result in sparser networks and more local dynamic patterns, while a larger time window size will average out small local variations and provides more stable trends. And the choice of the time window size should be based on the full time span of the data/event/network—a shorter time window size for more transient events and a longer time window size for more long-lasting ones. The type of network statistics to monitor is also an important factor to consider. For general purpose diffusion tracking, the ordinary ones would suffice: number of nodes and edges, average degree, or any other content related tweets statistics—fractions of retweets, URLs and hashtags etc. For more specialized patterns, such as connectedness, reciprocity, clustering or other path related information, I should resort to the corresponding specific statistics to handle. Also if I am more interested in how a single or a specific group of users behave during the diffusion instead of the global diffusion patterns, I should monitor the node level statistics—the type of centrality measure to monitor will subject to the requirements of the task: i.e. betweenness centrality for the "bridging" functionality of a node and PageRank for a general comprehensive measure of the node importance. Sometimes the scope of the underlying network is another factor to manipulate, especially when different levels of networks are already well defined. Just like the comparison I have shown in the Figure 2.4, different choices of networks can either enlighten or blur specific diffusion patterns, deciding which networks to choose will depend on the actual task to accomplish.

I have also shown that network statistics can not only be used to keep track of the event diffusion, but can also be used to distinguish event related hashtags versus non-event ones. First I used a figure to illustrate that an event hashtag can be visually differed from a non-event one using the network statistics evolution plots. Then I further investigated this hypothesis by performing K-means clustering and binary classification on selected hashtags from the 15k_user data sets. The results are not perfect, but they do demonstrate the network features have the ability to distinguish the event hashtags to some extent. However, due to data limitations, I will not be able to further verify this conclusion.

In summary, the network statistics can not only provide much insight to the dynamics of the diffusion, but can also reveal distinct properties of event related hashtags. But more data are needed to further validate the latter conclusion.

2.4.2 Future work

As notified previously, one main direction for future work would be to validate the obtained results for event hashtag identification. I need well defined classes or clusters of hashtags and accurate corresponding labels on each hashtag. Moreover, I need a large amount of data like this to drive more persuasive conclusion. Finding such data is not an easy task, much time and effort are needed, but the potential impact will also be great if I can show the network statistics can indeed generally distinguish hashtags of different types.

Another meaningful extension would be trying to build a real-time monitoring system to track diffusion process using various network statistics, on both a global and node level. Unlike the previous work described, this will be a task requiring intense engineering effort, but it is equally or even more impactful with practical implementation purposes.

Chapter 3: Predicting user influence during urgent event diffusion

3.1 Introduction

3.1.1 Motivation

The definition of a user's influence can vary in different contexts, and in my study I confine influence to be the ability to "spread information" during the urgent event diffusion on Twitter. Thus I choose the total sizes of retweet cascades generated by a given user to be the measure of the user influence. A retweet cascade can be simply considered as the full tree-like structure generated from the original tweet, where each edge is also labeled with a timestamp. Figure 3.1 shows some concrete examples of retweet cascades from the Sandy set. The majority of cascades will appear just like the upper one in the figure with centric star shapes; occasionally, there will be cascades with more complicated structures like ones in the lower part of the figure. Another part to clarify is that since I am measuring the user influence and a user can produce several popular tweets with different number of retweets, some weights are needed to aggregate these sizes of retweet cascades to generate a single score for that user. There are no well-established conclusions for this choice of weights from previous studies, therefore I will use the simplest method—the total size of all retweet cascades from a given user as the influence measure for that user. In this way, users with one highly popular cascade will be considered equally influential as ones with multiple median level cascades.

The chosen influence measure is only well defined for users who have posted some original tweets and at least received some retweets. I will refer to these users as seed users—implying they are the seeds of the diffusion. However, there are definitely many other users that also take part in the diffusion process, and some of them are also playing important roles—such as users who actively retweeting others. But for the purpose of this work, I will concentrate on the seed users. Thus I clarify here that the user influence here will literally imply the influence power resulting from the "seeding behavior", not something else.

Moreover, based on the literature review, I have discovered the following limitations on the influential node prediction:
1. Since the prediction problem is time sensitive, the authors will usually use data spanning a long period to train the model (Bakshy, 2011) (Galuba, 2010) (Kupavskii, 2012). However, this is not always feasible since the information given to me for training is often limited, especially for the urgent diffusion case—high diffusion rate, large diffusion scale and users will not be aware of the event in advance. In this situation the value of the prediction will largely depend on whether I can make decent

prediction when only having restricted information, which is often the case in the Twitter data due to various restrictions of Twitter API.

2. The network features derived from the actual diffusion networks are seldom used in prediction, often only the number of retweets are used; while features from users' social context--number of followers, number of friends or pagerank of F-F networks etc. have been used or suggested frequently (Bakshy, 2011) (Kupavskii, 2012) (Petrovic, 2011) (Hong, 2011) (Simmie, 2014). The structural and dynamic information contained in the diffusion network--the retweet network in my case, could potentially help the prediction.

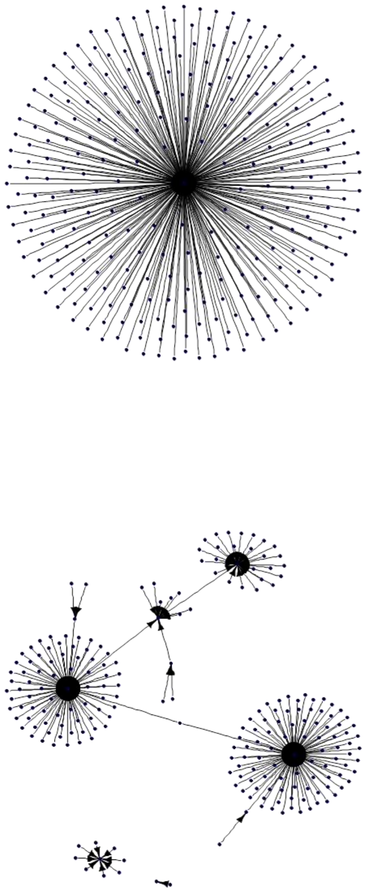


Figure 3.1 Example of retweet cascades from Sandy set

3. There is no a comprehensive comparison among different prediction methods, many authors have adopted decision tree based method—decision tree, random forest or gradient boosted tree. It is generally considered as a better approach than the classic GLM (generalized linear models) method (Bakshy, 2011) (Kupavskii, 2012),

but may not be the case everywhere; and the other state-of-art machine learning method that I will employ—deep learning has not yet been applied to this field.

In this Chapter, I attempt to work through the above limitations. I propose to select a specific number of top nodes based on given information, extract both baseline and network features (more details later) of these nodes, then implement three state-of-art machine learning methods to predict future influence scores for those nodes. By observing similar behavior among similar type of data sets(which will be discussed later), I create a feasible way to overcome the limited information by using past data sets as the training set to train the model instead of choosing both training and testing sets from the same data set. I will show this approach indeed produces decent results and that my methodology best suits the conditions of the Twitter streaming API, where I can collect real time streaming tweets with proper structural format about current major events using specific key words. All the features I have used in my models are directly included in those streaming tweets, and do not require any additional resources. I will show that addition of network statistics, especially dynamic network statistics can consistently produce significant improvement over baseline features on the prediction tasks with various experiment settings. Moreover, I also discover that the linear model consistently outperform the nonlinear ones in various settings, which could imply existence of strong linear relationship yet barely nonlinear relationship between the selected features and the response on the log scale (I perform log transformation on features and response before prediction).

3.1.2 Data description

Table 3.1 shows the comprehensive summary for the event data sets. The numbers within the parentheses are corresponding fractions. The retweet cascade depth is simply the length of the retweet chain, or the depth of the corresponding retweet tree. The four data sets differ on the scales, and the fractions of retweets and hashtags are increasing over time—representing the evolution of user habits and the increase in the Twitter use. Comparing the most two recent sets—Nepal and Blizzard, users seem to prefer adopting mentions (including retweets) and hashtags to spread information. Meanwhile less usage of links is observed in Blizzard set and this does not follow the trend in the rest three data sets where the fraction of links increased over time. The underlying reason for this is unknown—it could possibly just be normal variations among data sets. It is also observed that over time the retweet cascades are becoming "shallower"--the fraction of cascades with depth greater than 1 is dropping drastically. This perhaps also reflects changes in the user behavior—users are retweeting the source tweets directly instead of following the chains. The fraction of seed users, who are direct contributors of popular tweets, also decreased significantly. Both of these observations—"shallower" cascades and fewer seed users, seem to together imply that the general distribution of the retweets becomes more concentrated around a few seed users; considering this in concert with the trend in the increasing fraction of retweets, the chosen influence measure—total retweet cascades of seed users, becomes a more notable measure for user influence during the event diffusion.

Figure 3.2 includes the F-F distribution plots for several event sets under different conditions. All the plots are restricted to be within one week from the starting date of the data collection for equal comparison. Despite of this, there are still too many users to be shown in one plot, the users are further truncated by their influence scores—the total size of retweet cascades. The threshold of the scores I will consider is labeled as TH.

Data set	**Irene**	**Sandy**	**Nepal**	**Blizzard**
No. tweets	3178604	16529593	7310929	1716234
No. users	1487056	5588298	2229702	941488
No. retweets	965158(0.304)	8410769(0.509)	3905264(0.534)	1101108(0.642)
No. retweet cascades with depth>1	71684(0.074)	408919(0.049)	27716(0.007)	14644(0.008)
No. mentions	1874213(0.590)	9272919(0.561)	4381043(0.599)	1226905(0.715)
No. tweets with URLs	395835(0.124)	5394189(0.326)	4562842(0.624)	643483(0.375)
No. tweets with hashtags	102954(0.032)	7393790(0.447)	2959370(0.405)	1369829(0.798)
No. seed users	218298(0.147)	981626(0.176)	184863(0.083)	62611(0.067)
Time frame	26 Aug 2011 14:00:00 GMT ---- 12 Sep 2011 23:31:26 GMT	25 Oct 2012 04:00:00 GMT --- 06 Nov 2012 04:59:59 GMT	27 Apr 2015 02:00:02 GMT -- 22 May 2015 00:24:45 GMT	22 Jan 2016 16:09:42 GMT – 31 Jan 2016 05:23:00 GMT

Table 3.1 Event data sets comprehensive summary

The upper two plots are from the Nepal set with TH=100 and 500 respectively, and the lower two plots come from the Irene and Sandy set with TH=500. The rough shape of the slopes is similar to those displayed in Chapter 2. Also users with a higher number of followers constitute a higher fraction with increasing TH, which is natural since the higher number of followers implies a higher potential of being retweeted.

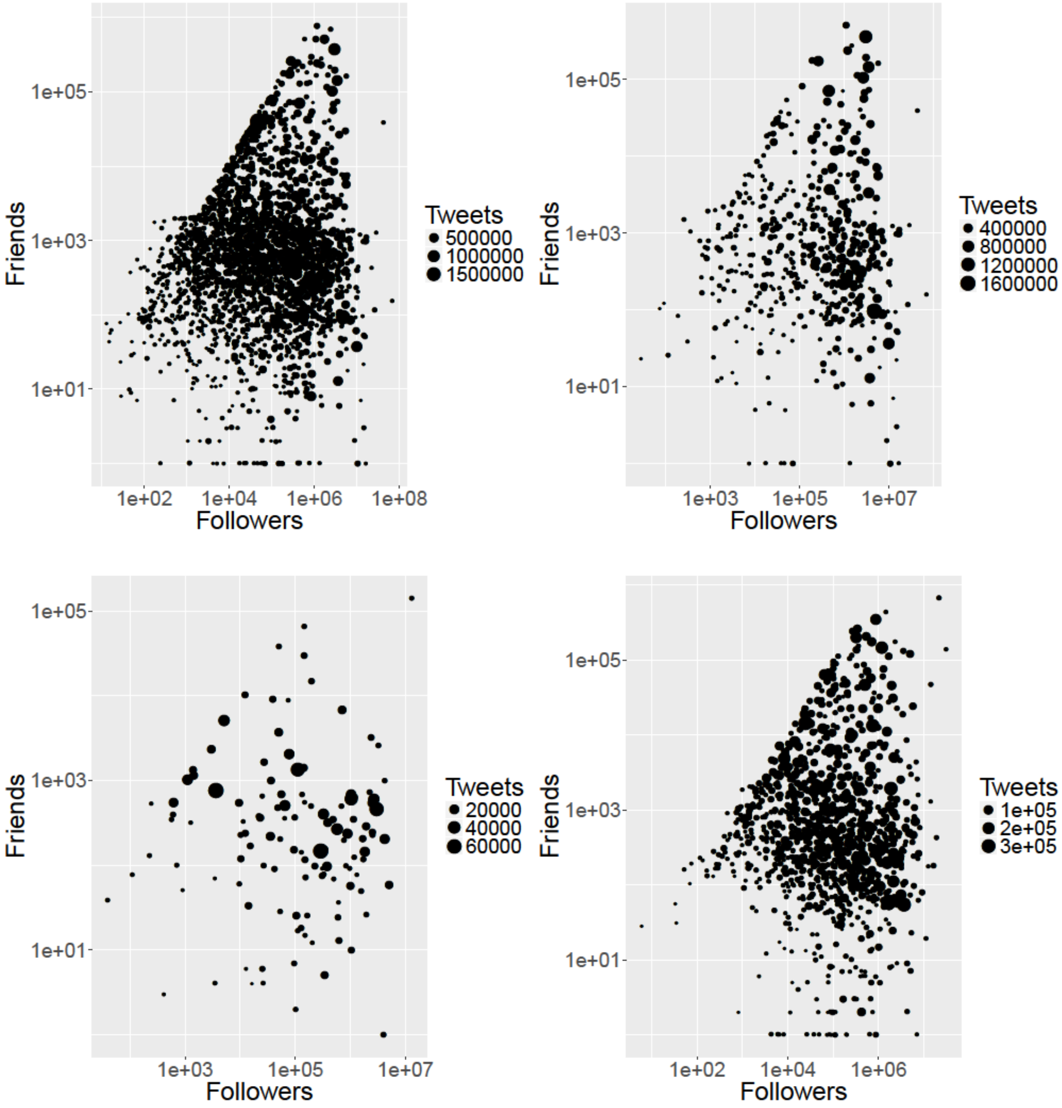
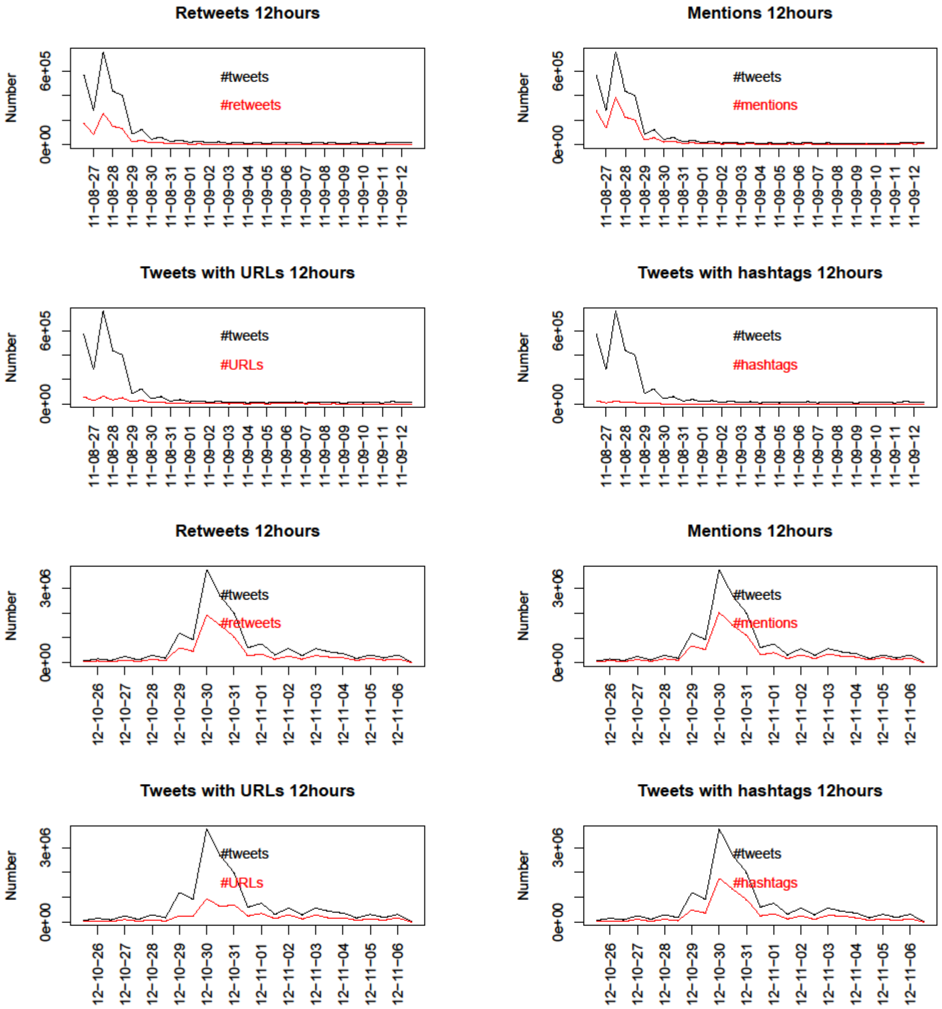


Figure 3.2 F-F distribution plots for event data sets (the upper two plots are from the Nepal set with TH=100 and 500 respectively; the lower two are plots from Irene set and Sandy set with TH=500 respectively)

Comparing the three plots with TH=500, the differences on the data scale also are also reflected here: the Sandy set has the densest plot and the Irene set has the fewest users in the plot. Also I find the users seem to be a little concentrated round the band with the number of friends between 100 and 1000, meaning higher number of friends does not necessarily lead to more retweets.

Figure 3.3 is a collection of evolution plots for each of the four event data sets. Except for Sandy set, the other three sets more or less start with the peak period, this is inevitable for urgent event diffusion since the data collection only begins after the start of the event and it is not possible to anticipate the occurrence of the event before it actually happens.



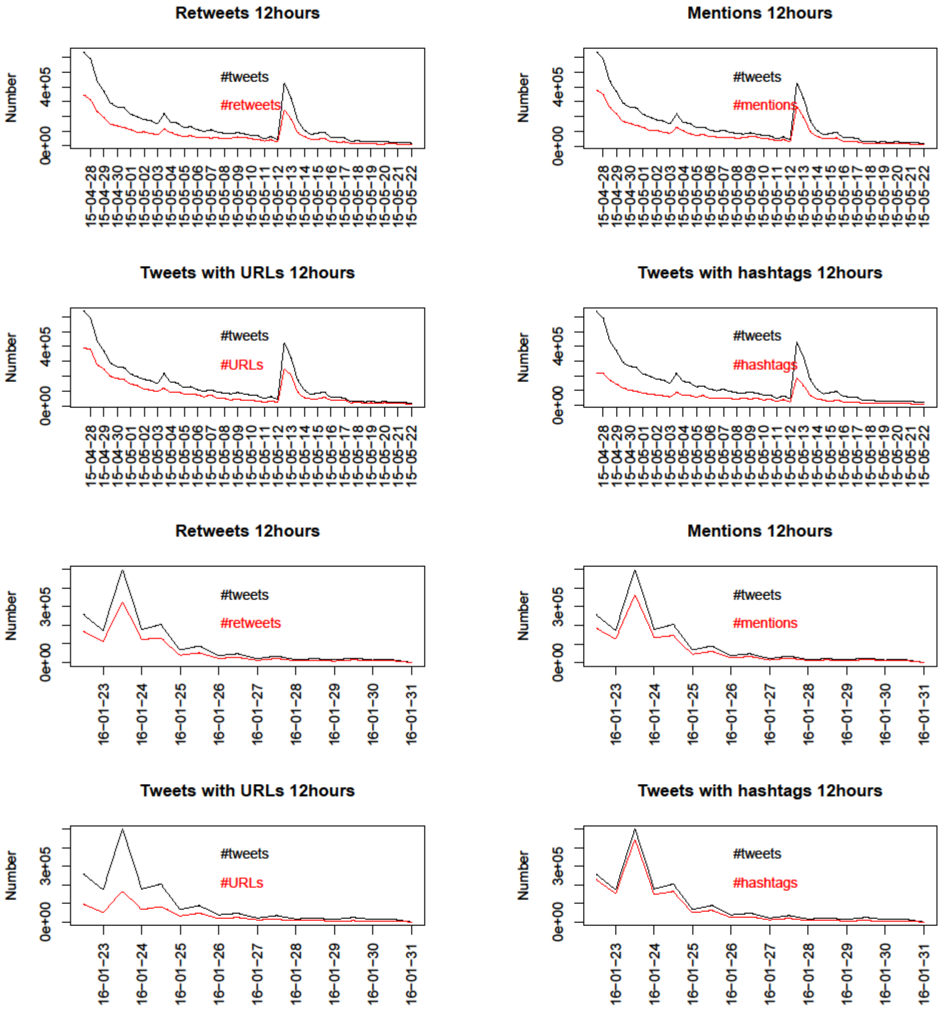


Figure 3.3 Evolution plots for event data sets (from top to bottom, the plots are for Irene, Sandy, Nepal and Blizzard set respectively)

The plots show different components of tweets are evolving with similar a pace with the total number of tweets. Across data sets from earlier timestamps to later timestamps, hashtags and URLs are adopted more frequently; while mentions and retweets also have a small increase but remain relatively stable. All these observations coincide with the data summary table. The Sandy set seems a little special with peak occurs several days after the data collection. After investigation, the peak time seems to correspond to the landing time of Hurricane Sandy onto the eastern shore of US. Since Hurricane Sandy started at Caribbean Sea instead of US mainland, thus the discussion began earlier than it actually landed onto US. To identify how this delaying peak time will affect my prediction, I created another data set called Sandy 4.5 for direct comparison, which is simply defined as a truncated version of Sandy set by starting 4.5 days later than the starting time of data collection.

3.2 Prediction task formulation

3.2.1 Task types

Generally speaking, I have formulated three prediction tasks: rank prediction, classification and direct prediction, they are distinguished by types of the response variables. Direct prediction simply means predicting the users' raw scores—total size of retweet cascades. Rank prediction only cares about predicting the ranking of the users instead of their actual scores. Classification will aim to identify a specific class of users. There are two types of classification tasks performed in my study: top quantile users classification—e.g. binary classification on top 5% users by the scores; and rapidly increasing users classification (only users with response exceeding a certain threshold are selected)—e.g. binary classification on users who have scores increasing greater than 5 times but also with scores greater than 10.

The rank prediction task will be my main focus, due to the following considerations:
1. Due to the limited information dilemma, I adopt the "cross prediction" mode—using data sets corresponding to past events as training data to predict future events sets. However, different data sets have different scales. Thus it is a better choice to standardize (subtract the mean and divide by the standard deviation) the data sets before performing the prediction, which means the prediction scores will be in normalized forms and cannot recover to the raw cascades scores. Under this condition, what I obtain from the models will be the normalized scores, which implies that I am actually performing a rank prediction and not a direct quantity prediction.

2. Since standardization has been performed, the results obtained from rank prediction will usually be more reliable and stable than direct prediction; this could still be the case even when no heterogeneity exists among the data sets.

3. It will be shown that the classification task can actually be accomplished together with the rank prediction task.

As to the performance measure, I will use Spearman rank correlation for rank prediction (the results are similar if using Kendall Tau correlation, another frequently used metric for ranks). For classification, I will use the area under the precision-recall curve (PRAUC), which proves to be an informative measure for highly imbalanced classification tasks. As to the direct prediction, I simply use the root mean square error (RMSE). The rank correlation is computed using the R base function "cor" and rank of ties is defined as the average. The PRAUC is computed using the R package PRROC, so does the plot of the PR curve

Even though I have put much emphasis on rank prediction, I will display results from all three types of prediction for comparison. And which task to perform in practice is subject to what kind of problem needs to be solved.

3.2.2 Features

The features used in all three tasks will be the same, they are all user-specific features and include two categories—baseline features and network features, and network features are consist of static network features and dynamic network features:
1. The baseline features are features often used in previous prediction studies yet also available in my data, they are:
'Total of tweets', 'Number of followers', 'Number of friends', 'Number of tweets', 'Number of mentions', 'Number of tweets with URLs', 'Number of tweets with hashtags', and 'Past scores'. Past scores are users influence scores within the feature collection period, while the prediction responses are influence scores within the prediction period (more details mentioned later). The total tweets are the total number of tweets posted by the given user since creation of the account; and number of tweets only count the tweets posted from the beginning of data collection, this is the same for other features from various specific types of tweets.

2. The network features are constructed using the following network statistics: 'Indegree', 'Outdegree', 'Pagerank', 'Authority score', 'Hub score', Eigenvector centrality', 'Closeness centrality' and 'Local transitivity' (local clustering coefficient).
(i) The static network features are aggregate statistics of the whole network within the feature collection period.
(ii) The dynamic network features are ones from subnetworks within certain time interval. Compared to the static features, dynamic features contain rich dynamic information and they could provide additional improvement for the prediction.
Here I am not including betweenness centrality due to its extremely high computation cost and limited benefit.

The features mentioned above are almost all of the features that can be derived from the data; some other commonly used features, such as network features derived from the F-F networks, are not available in my data.

3.2.3 Training and testing sets

As mentioned before, training sets will be data sets concerning to past events compared to the selected testing set. For example, majority of my analysis will use Nepal set as the testing set, and the default training set will be Irene set combined with Sandy set; results from Irene set alone or Sandy set alone as training sets are listed for comparison. Predictions with testing set Sandy or Blizzard are performed to validate the conclusion, and the default training sets are Irene and Irene+Sandy respectively. It is worthwhile to note the standardization for rank prediction should be performed within each data set separately if multiple training sets are involved, this is still due to the scale differences among data sets.

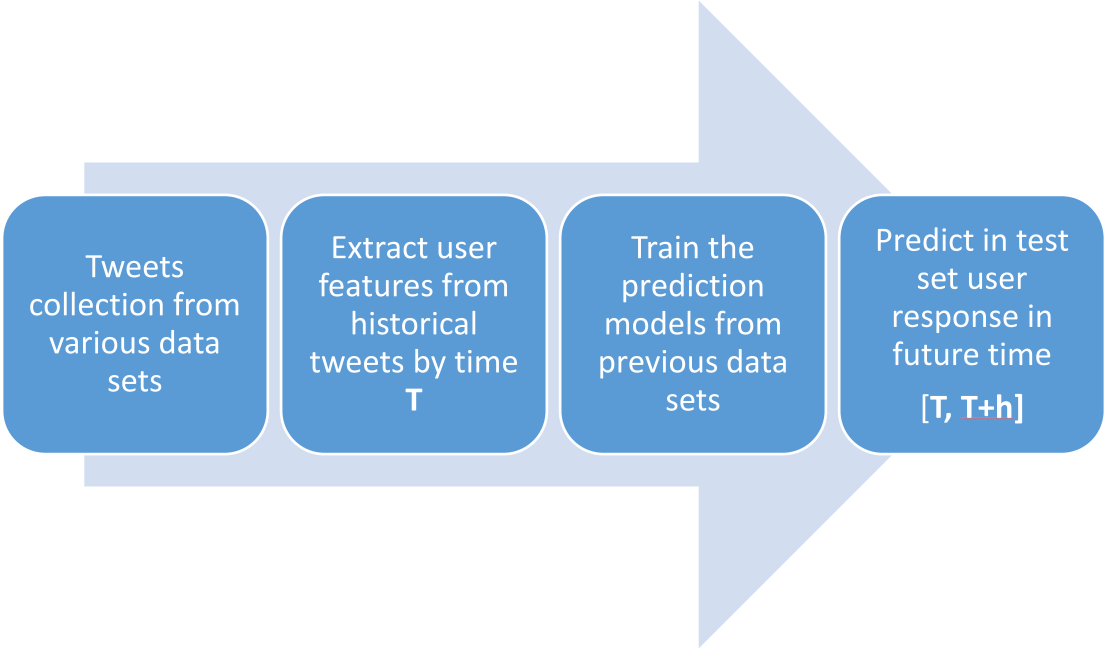

Tweets collection from various data sets
Extract user features from historical tweets by time T
Train the prediction models from previous data sets
Predict in test set user response in future time
[T, T+h]

Figure 3.4 A general prediction task flowchart

What I would like to obtain from these settings is how the choice of training sets and testing sets will affect the prediction results. It turns out they are one of the most deciding factors on the prediction performance. Especially when the testing set is usually fixed in practical applications, choice of training sets from a list of potential candidates will be extremely important, and I will develop further discussion on this in later sections.

3.2.4 Task parameters

In my prediction study, the following parameters are used:

T: The feature collection period length, from period [0, T] I can construct the whole network and extract both static and dynamic network features.
h: The prediction period, indicating the length of period for future user score prediction. Thus the responses for my experiments will be cascades size within period [T,T+h] for nodes under consideration.
Δ: Time window size for dynamic network features extraction, use segment statistics series to reduce features correlations.
k: Size of top nodes (ranked by past scores) to monitor, the upper bound is 10k in my case.

The general flowchart for the prediction task is shown in Figure 3.4. The default values for the parameters are: T=1d, h=5d, Δ=6h and k=10000; I will stick to these values unless special notes are informed.

Figure 3.5 shows the response distribution among various data sets with the default parameter settings. The first one is the normal log-log distribution plot and the second one is the CDF (cumulative distribution function) plot of standardized score. From the log-log plot I can clearly see the distribution fits the power law distribution well. The normalized CDF plot demonstrates using normalization (standardization) can indeed reduce heterogeneity among data sets; although the distribution differences are still obvious after normalization—especially for nodes with lower scores, but this may not be as serious as it appears since I care more about top-ranked nodes, where the distributions are much closer from each other.

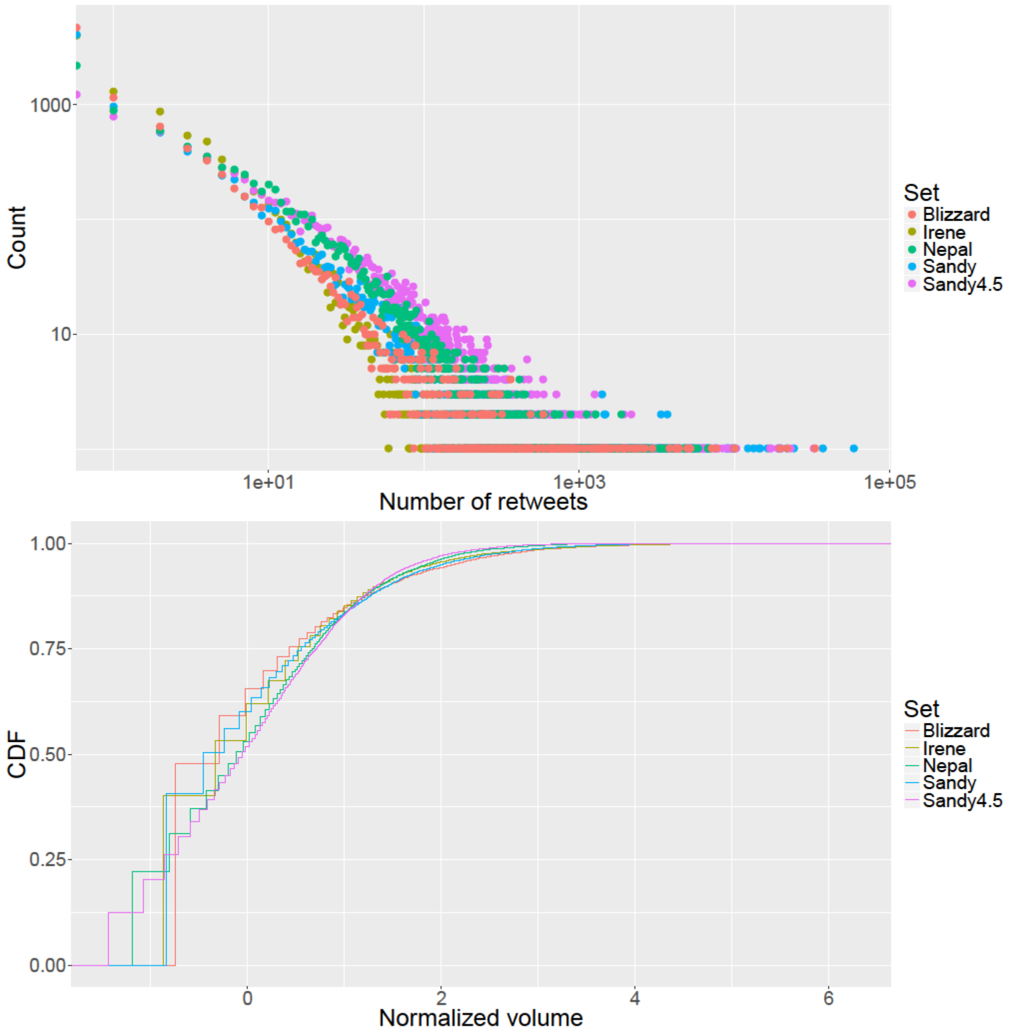


Figure 3.5 User scores distribution with various data sets with default parameters

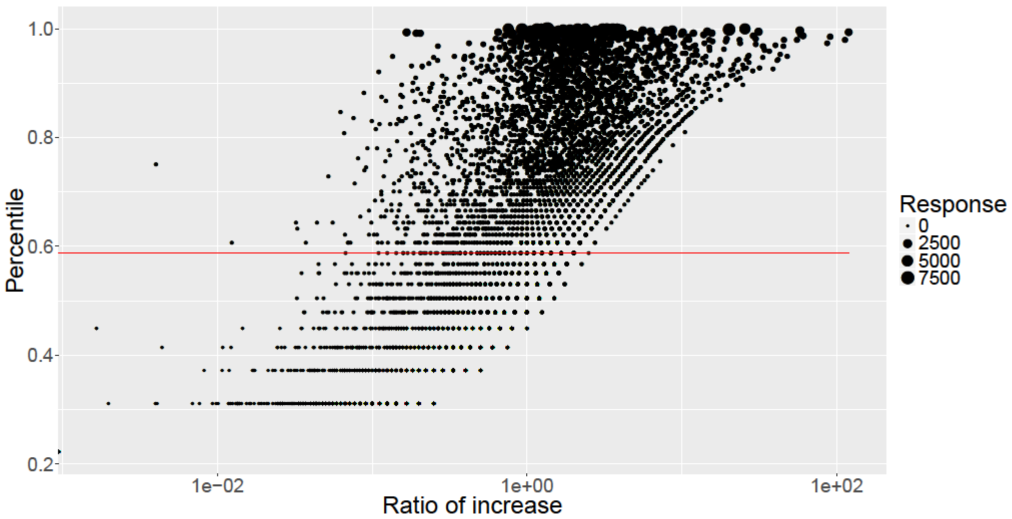


Figure 3.6 User scores percentile and ratio of increase distribution for Nepal set with default parameters

Figure 3.6 shows the response distribution for Nepal set in another aspect. The y-axis is simply the quantile (percentile) of the response; the x-axis is the ratio of increase of the influence scores, which is defined as the ratio of response score (cascades size during [T, T+h]) over past scores (cascades size during [0,T]). This ratio is the key factor to perform the classification on rapidly increasing users. The actual response size is illustrated by size of the point. And the red horizontal line represents the cutoff value to truncate rapidly increasing users—set to be 10 as the default, which corresponds to about 0.6 in quantile. Thus for the top quantile classification task, my main focus is on top bands of the plot; while for rapidly increasing classification task, my main focus is on nodes staying in the top right corner.

Table 3.2 shows how the nodes response distribution changes over T among various sets, notice only nodes who have shown up in the retweet network (either retweet others or being retweeted at least once) are counted. The columns from top to down are: total nodes count, nodes response 75% quantile, nodes response 95% quantile, nodes count with response >=10, nodes count with response >=100, numbers in the parentheses are corresponding counting fractions within top 10000 nodes (ranked by past scores), which can be used to illustrate how representative the top 10k nodes are. The values in the first row are just the varying T values and h is always equal to the default value 5d. I can see from the table if I do not restrict the number of nodes to the top 10k, the response distribution will be more skewed. This implies that the majority of involved nodes are completely inactive in the next 5 days, they may just retweet others' tweets but they do not contribute the content for retweeting. Also the top 10k nodes ranked by their past scores can constitute majority of important nodes, although there are always missing ones since the base number of nodes is too large. Thus choosing the top 10k nodes would suit the task. On one hand, extracting features from all nodes would be unnecessary and waste time; on the other hand, even given sufficient computational power, it will do no good for the prediction since it is simply adding noise to the data by adding data points with null values. In conclusion, all the prediction tasks in this study will be confined to the top 10k nodes ranked by past scores—e.g. the rank or the quantile will refer the rank or quantile within the top 10k nodes.

		1d	2d	3d	4d	5d
Irene	**Total**	235870	445808	581597	615176	630557
	75%	0	0	0	0	0
	95%	3	1	0	0	0
	>=10	3047(0.529)	1324(0.654)	538(0.766)	342(0.795)	232(0.810)
	>=100	230(0.813)	128(0.891)	55(0.909)	34(0.882)	24(0.917)
Sandy	**Total**	74232	170010	291564	716825	1963399
	75%	0	0	0	0	0
	95%	13	10	8	5	2
	>=10	4494(0.620)	9060(0.443)	13307(0.355)	21045(0.280)	23765(0.275)
	>=100	952(0.808)	1603(0.750)	2071(0.704)	2895(0.684)	3065(0.727)
Nepal	**Total**	427359	652866	797268	899826	979524
	75%	0	0	0	0	0
	95%	1	0	0	0	0
	>=10	6482(0.666)	6341(0.625)	5897(0.632)	5374(0.631)	4979(0.636)
	>=100	1130(0.911)	1078(0.869)	1008(0.856)	921(0.847)	872(0.841)
Blizzard	**Total**	214508	497547	602503	647983	672856
	75%	0	0	0	0	0
	95%	0	0	0	0	0
	>=10	2384(0.778)	1874(0.820)	1140(0.814)	753(0.841)	476(0.840)
	>=100	461(0.887)	272(0.941)	137(0.905)	88(0.898)	58(0.914)

Table 3.2 User response distribution over varying T among various data sets (h=5d)

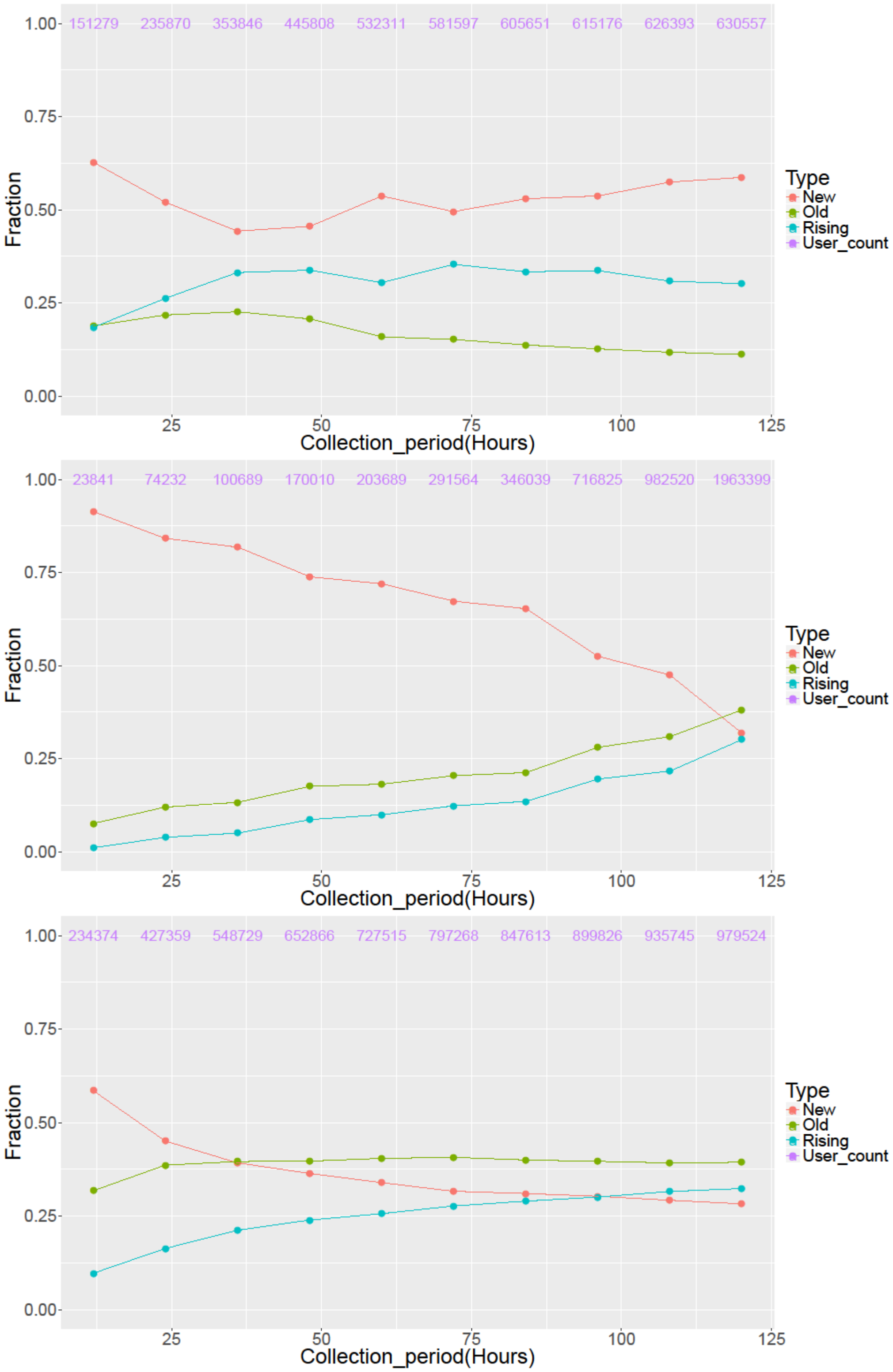


Figure 3.7 Top 10k user components analysis over varying T values (h=5d) for Irene set, Sandy set and Nepal set from top to bottom respectively

The other information I can obtain from the table is the overall trend of activity by looking at the counts of top nodes. For example, the number of nodes with response greater than 10 has decreased from around 3k to only about 200, this is indicating the diffusion is gradually dying out. A similar trend is observed in Nepal set and a reverse trend is observed in Sandy set, which coincides with what see from the evolution plots.

With the nodes selection, if performing a rank prediction the rank will simply be based on the selected nodes. This is inevitable unless no nodes selection is performed, which is not suggested due to previous discussion. There could be complicated trade-off considerations on what is the best k value to choose, but this is out of scope of my discussion and I will just stick to k=10k in my study.

Figure 3.7 demonstrates the other aspect of dynamics of nodes response distribution—what the real top 10k nodes by response consist of. The three plots correspond to Irene set, Sandy set and Nepal set from top to bottom respectively. The purple numbers on top of plots are total nodes count by the given T values. There are three types of nodes involved in the plots—new nodes, old nodes and rising nodes. There are defined as follows:
1. Old nodes: these are nodes belonging to the top 10k nodes in the past (within period [0, T])
2. Rising nodes: these are nodes showing up in the past but are not among the top 10k nodes.
3. New nodes: these are nodes not showing up in the past at all, they simply emerge within period [T, T+h].

The y-axis in the plots displays the fractions of each type of nodes over top 10k nodes ranked by their responses. Fractions of different types of nodes can shed light on the underlying diffusion dynamics. In the Irene plot, new nodes and rising nodes are both staying above and remain stable, which implies: there are consistently new users joining the conversation and generating "impact"; there are not many old users continuously remain active (less than 20% and still decreasing over time), meaning the ranking is not stable. In the Sandy plot, the trend is clear: the new users are dropping over time, old and rising users are gradually catching up. If I refer to the evolution plot of Sandy set, I will find around 5 days from starting time the diffusion reaches peak hour, which just corresponds to the tails of curves. From there fraction of old nodes begin taking the lead—showing the ranking is becoming more stable from then on. Similar patterns are observed in the Nepal set, where old nodes are taking relatively steady fractions over time—implying stable ranking. And new nodes fractions are dropping—implying the diffusion are becoming saturated. Thus I can see rich information about the dynamics of the diffusion can be derived from the plots.

3.2.5 Methods

I adopt three state-of-art machine learning methods for my prediction tasks: LASSO linear regression (belongs to one type of generalized linear model, will simply call this GLM from now on), random forest and deep learning (deep neural network). While GLM consists of nearly all popular linear models of various types, the other two methods are typical non-linear models.

$$\min_{\beta \in R^n} \left[\frac{1}{2l} \sum_{i=1}^{l} (y_i - x_i^T \cdot \beta)^2 + \lambda P_\alpha(\beta) \right]$$

$$P_\alpha(\beta) = (1 - \alpha) \frac{1}{2} \|\beta\|^2 + \alpha \|\beta\|$$

The above equations show the loss function of elastic net, which is a more general model than LASSO regression. It still has the same square loss function as the ordinary linear regression, but with addition of a penalized term to regularize magnitude of predictors to prevent overfitting (Babyak, 2004). The penalized term for the elastic net model is the weighted sum of L1 norm and L2 norm of the features as above. When α=0 the model will become ridge regression and when α=1 it will become LASSO regression. The lower plot illustrates how LASSO regression works differently as opposed to ridge regression. In the vector space formed by the parameters, the optimal parameters to be estimated will be the intersection of the contour plot of the square loss function and the geometry object formed by the penalized term. This geometry object formed by L1 penalty term and L2 penalty term are hyper-cube and hyper-sphere in the parameter space respectively. Since the contour plot of the loss function is convex, the intersection with L1 penalty term will inevitably be at the vertices or edges of the hyper-cube. This will force some parameters values to be zeroes, and thus the features selection is achieved. Therefore LASSO regression is a linear model that can prevent over fitting and achieve variable selection, and is used as the deputy of linear model in my task.

Random forest is an ensemble method of decision (regression) trees. Decision trees are a popular method used in a previous study of social media influence (Bakshy, 2011) (Kupavskii, 2012): the basic idea is to perform recursive splitting of features values to grouping data points to various leaves based on certain criteria. The features and response values within each leaf will be used to decide the final output; the convention is majority votes for classification and simple average for real-valued prediction. While the random forest method will collect results from many decision trees and aggregate them together to produce the final output. Each tree in a random forest is formed by splitting a group of randomly selected features instead of all features based on a random sample of full data, and these are where the term "random" come from. All these can help reduce correlation among trees so as to avoid overfitting. Generally speaking, random forest is a powerful nonparametric yet nonlinear model, but the drawback will be the poor model interpretability compared to linear models.

Another state-of-art nonlinear machine learning method is deep learning or a deep neural network. Although there has been debate about how "deep" a neural network should be in order to call it deep learning, I will skip this discussion and simply follows the convention from

Wikipedia(https://en.wikipedia.org/wiki/Deep_learning)–network with hidden layers greater or equal than two is considered as deep learning.

For the deep neural network the features values are fed as input layer, the values from previous layer of neural network will experience similar types of nonlinear transformation to generate values for the next layer. The transformation will usually be to multiply by some weights and then applied to certain pre-defined activation function—e.g. the sigmoid function and the tanh function. The choice of activation function will generally be subject to the field of the application, but sometimes a different activation function is also needed for the output layer depending on the type of target response. Overfitting can be avoided either by setting a dropout ratio to randomly remove a certain fraction of values fed to the next layer or by adding penalized terms just like in GLM. Deep learning is a powerful machine learning method and is actually the only effective method for certain tasks (LeCun, 2015). This power could be explained by the Universal Approximation Theorem (Hornik, 1991), which provides a theoretical foundation the mechanism of the method. Moreover, there has not been a comprehensive application of deep learning to social media user predictive analysis, thus I choose this approach in my study to see its performance for my tasks.

For the implementation, I perform all my prediction analysis using R. The packages I have used include: igraph for network construction and features extraction, glmnet, glm2, party and h2o for the prediction. I adopt the default settings for all the three methods:

1. For LASSO GLM, I use the 10-fold cross validation (CV) and refit the model with selected features that gives the least CV error.

2. For random forest, I reduce the number of trees from 500 to 200 for runtime consideration. Each tree is trained on an independent bootstrap sample of the training data with a default number of 5 features randomly selected for each splitting. The final prediction outputs are the average or majority votes of all trees.

3. For deep learning, I use 2 hidden layers of 200 neurons in each layer, with hyperbolic tangent activation function, quadratic loss function for real-valued prediction and cross entropy loss function for classification, with the input features dropout ratio as 0.5 within each hidden layer.

A brief tuning parameters search was completed but the above default settings proved the best, thus I will adopt them in my experiments. More careful grid searches can be carried out to find optimum settings for each method for particular implementations.

For rank prediction and classification, all the features and responses are first log transformed and then standardized (centered by means and scaled by standard deviations) within the nodes selected (10k in my case). For direct prediction, they are just log transformed and not standardized. For the classification experiments, down-sampling is performed to obtain balanced training set and average prediction scores are used to produce the PRAUC values.

3.2.6 Variable importance

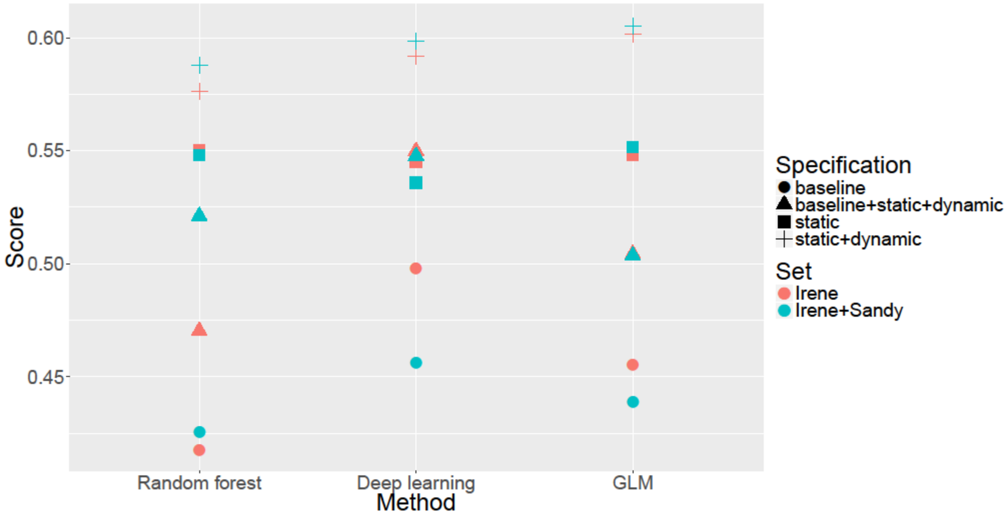
Variable importance is another useful output of the prediction other than the predicted values. It reflects how effective a given predictor on the given response when the predictor is not well correlated with other predictors. Among the three methods I have used, only GLM provides a rigorous defined yet easily interpreted variable importance measure. This is one of the biggest advantages of the linear model. Although not perfect, there are some usable variable importance metrics for the rest two methods. The variable importance option provided for random forest is obtained by permuting features values and monitor how the prediction performance varies with it. If the values of more important features are permuted, the performance will suffer more serious degradation, thus a variable importance measure can be computed based on this. For deep learning, there is an approximate measure to aggregating weights of first few layers of the neural networks, this may work well for shallow neural networks, but for really "deep" ones there are still no proper candidates.

3.3 Prediction results

3.3.1 Rank prediction

Now I show the results of rank prediction with various parameters settings, but as mentioned earlier, I will always stick to the default settings unless specific notifications. All the rank correlation scores are obtained from the average prediction scores out of n=10 (as default value) replicated runs of the experiment.

Figure 3.8 shows the rank prediction results for Nepal set varying methods and training sets. In the first plot, the differences among methods are subtle, GLM performs slightly better. The feature combination static+dynamic delivers the best results, compared to results with static only, I can tell dynamic features indeed help the prediction much. Also addition of network features to baseline ones indeed improvement the performance significantly. Since the two nonlinear methods fail to beat GLM, I will adopt GLM as the default method from now on since it is fast to run and straightforward to interpret. The next two plots show the results from GLM with varying number of nodes involved. The x-axis labels the number of top nodes included for rank correlation calculation, of which are ranked by past scores; the corresponding y axis is the rank correlation measured within that group of nodes. Therefore it is not surprising to see the results exhibit much more variations for fewer nodes and stabilize when the number is large enough. The network features, especially the dynamic ones, outperform the baseline ones much more with smaller nodes count than larger nodes count. I believe this is because the top nodes (top 10 or top 50), compared to lower ranked ones, have exhibited rich activity and thus have generated informative network features. Thus the functionality of the network statistics can be maximally explored; with more and more nodes involved—especially those much less active and lower ranked nodes, the predicting power of network statistics will decrease gradually since many of them will be almost inactive or even have null values. This is not the case for some of the baseline features, such as number of followers/friends and total of tweets, which will remain relatively constant regardless nodes activity, thus baseline features will generally provide the most stable (although not the best) results. Addition of network features consistent outperform baseline ones with changing number of nodes count, and for Nepal set even network features alone can provide much better results. Also the choice of training sets here does not matter much—they all produce comparable results. Irene+Sandy does a slightly better job and thus will be used as the default training set.



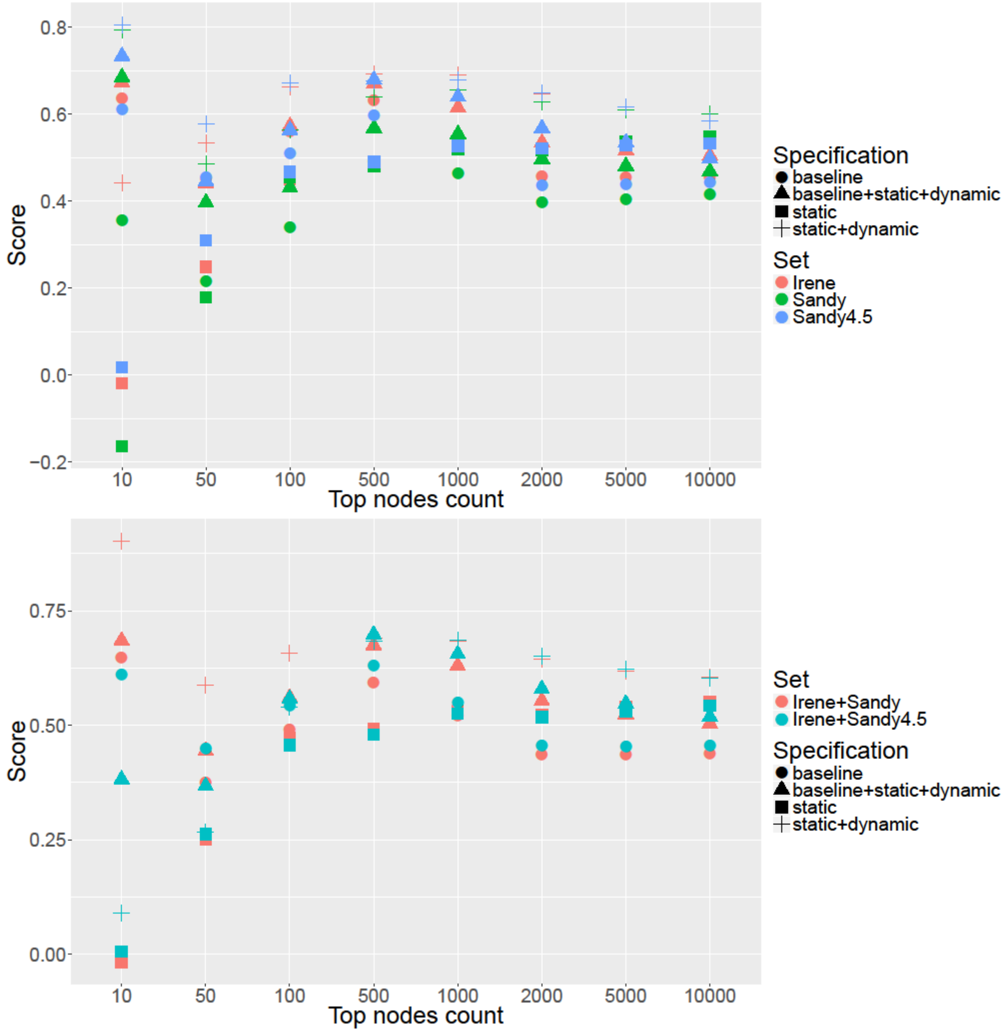


Figure 3.8 Rank predictions using GLM on Nepal set varying methods and training sets

Figure 3.9 displays the rank prediction results of Nepal set varying T, h and Δ (all default settings are applied unless otherwise specified). The titles of the first four plots simply represent the four different values of Ts. The addition of network features can consistently improve the performance under various settings, though the extent of improvement is different depending on the conditions. The best on average feature combination across all settings still seems to be static+dynamic, showing the network features alone can have good predicting power, at least to the Nepal testing set. While I do not see significant performance differences with varying T and h, Δ with 6h indeed does obviously better than Δ with 3h. Considering the static feature combination as the case when Δ=24h, there seems to exist an optimum value for Δ to give the best performance. A possible explanation for this is: too large Δ cannot provide enough dynamic information while too small Δ value may incorporate many null feature points thus also degrade the performance due to overfitting. This optimum value will depend heavily on the training and testing sets and can be found with fine tuning carefully, though the current settings seem satisfactory.

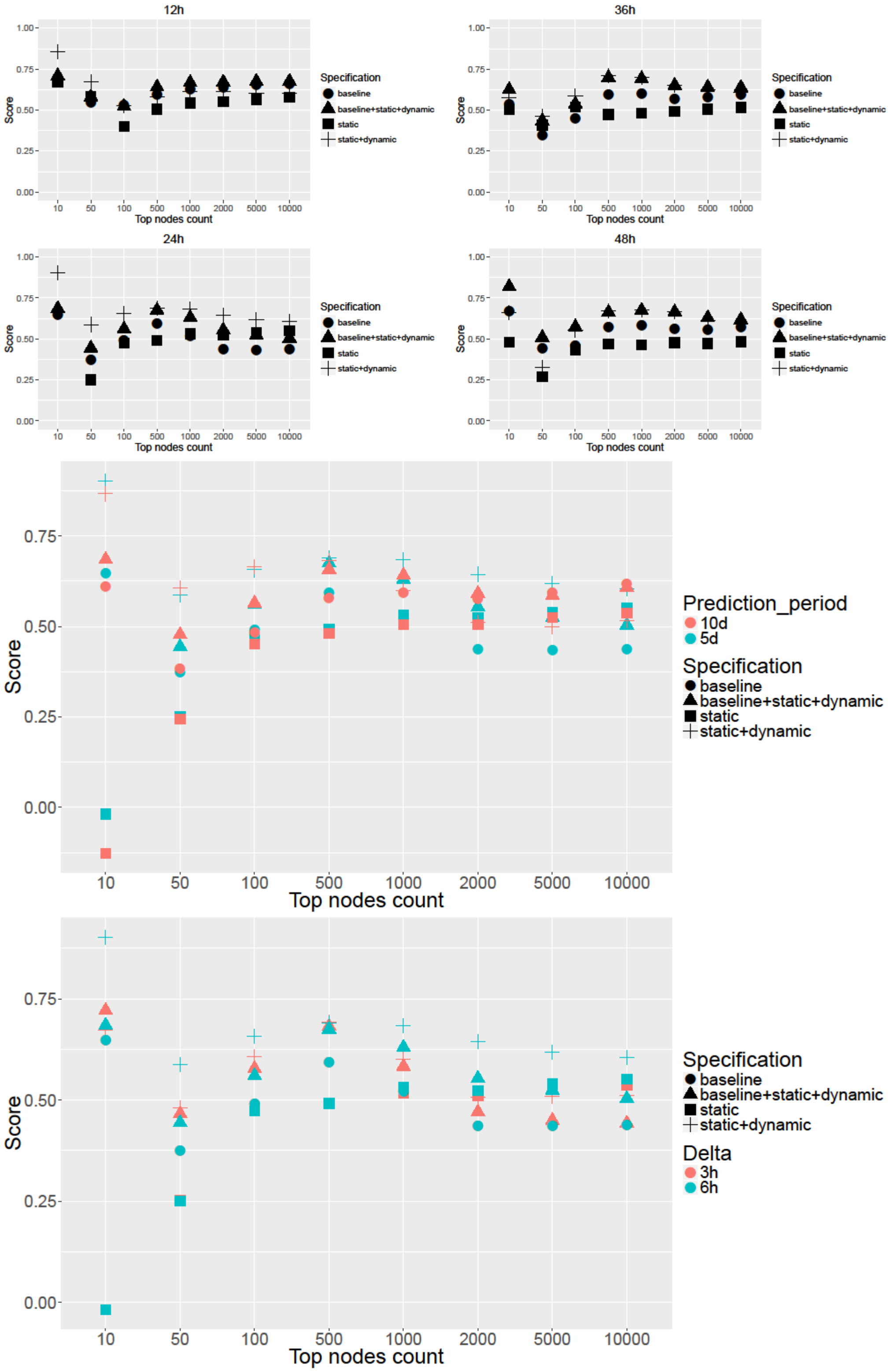


Figure 3.9 Rank prediction using GLM on Nepal set varying T, h and Δ

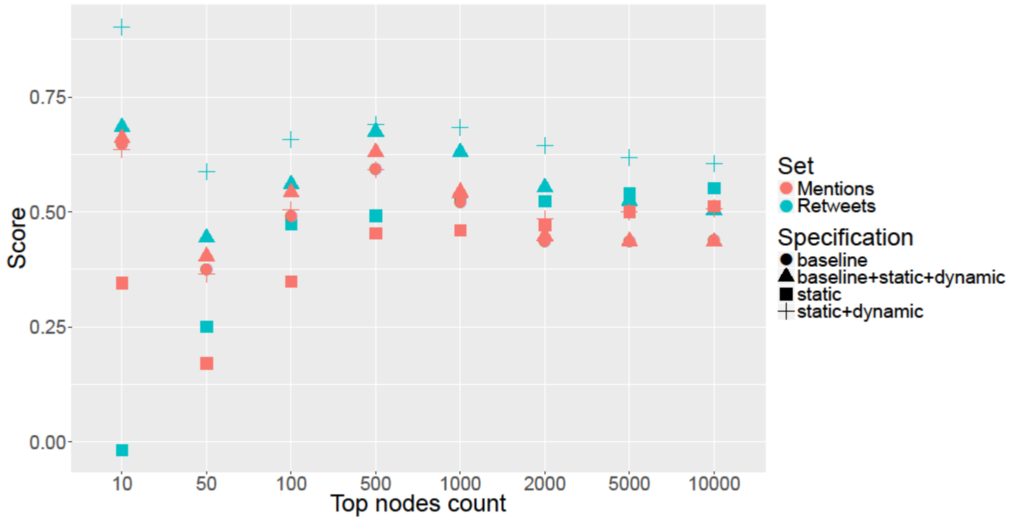
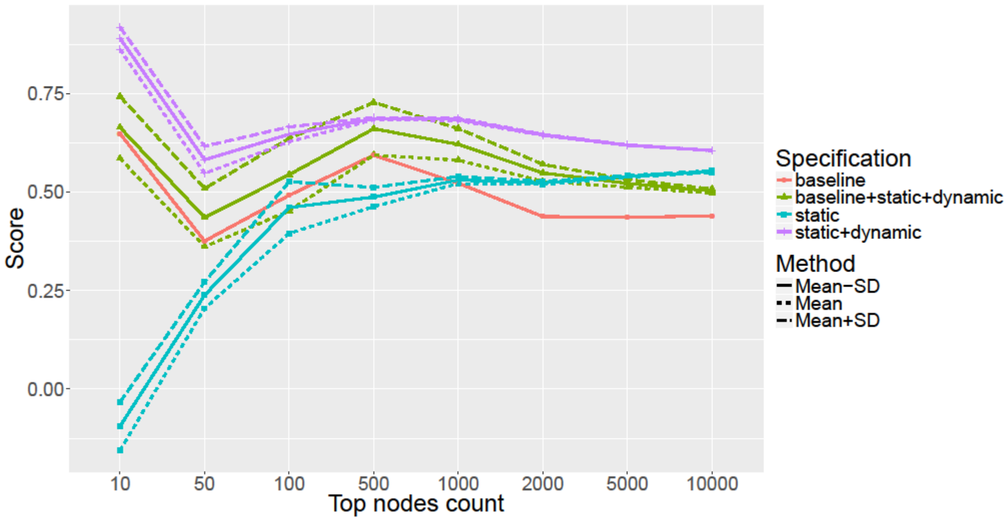


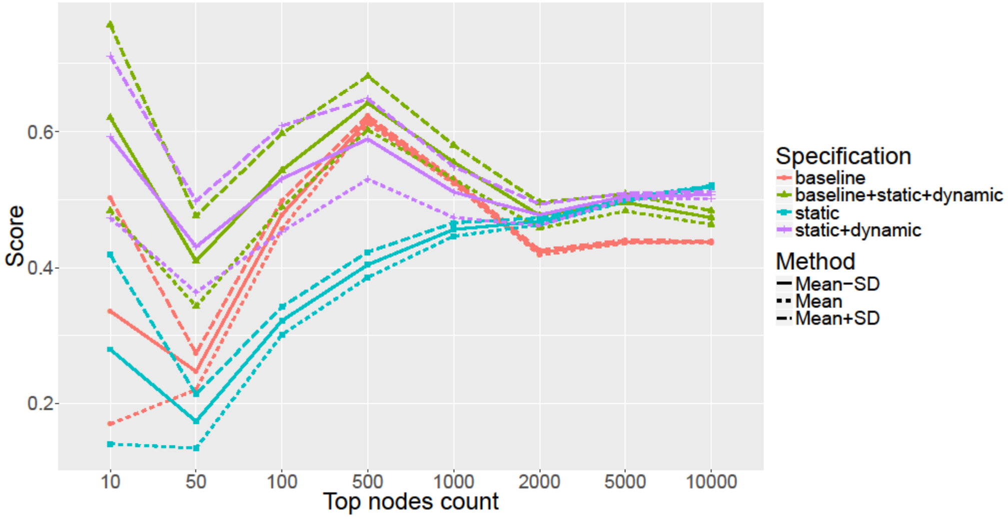
Figure 3.10 Rank prediction using GLM on Nepal set varying the underlying networks

The above Figure 3.10 compares the rank prediction results using two network statistics from two different networks—retweet network and mention network. In Twitter the mention network is implied denoted with "@" and it is common that a user can mention multiple users in a single tweet, which is unlike retweets. And all retweets generally belong to mentions, so do all replies towards the tweets. Thus I can consider the mention network is including all sorts of user communication, but is not mainly diffusion-oriented. Therefore retweet network features outperform mention ones significantly, though the mention network will be much better connected than the retweet network.

I have already shown the network features can improve the prediction performance with respect to the baseline ones, but I would like to further investigate whether this improvement is stable or not. Thus I carry out the stability analysis in Figure 3.11. The rank correlation is calculated from each single run of the experiment instead of the average scores from the above plots. Then a rough distribution of the rank correlation scores is simulated by repeating the experiments n=100 times. The mean and mean ±sd values are computed for comparison. For GLM both static+dynamic and baseline+static+dynamic outperform the baseline statistically. Moreover, GLM stands out the rest two in terms of stability, which is not surprising since both the rest two methods have much "randomness" associated with them, especially for deep learning due to hidden layer values drop out.

Figure 3.12 and Figure 3.13 show the rank prediction results for Sandy set and Blizzard set as the extension study. The effect of the testing sets is clearly large, especially for the network features. The best feature combination is baseline+staic+dynamic and the improvement towards baseline is less. This implies testing sets will have considerable influence on the predicting power of network statistics, but less influence for the baseline ones since they remain stable over time. That might be the reason why many of the baseline features share popularity among previous related studies—they do provide robust results that are less affected by the data sets under study.





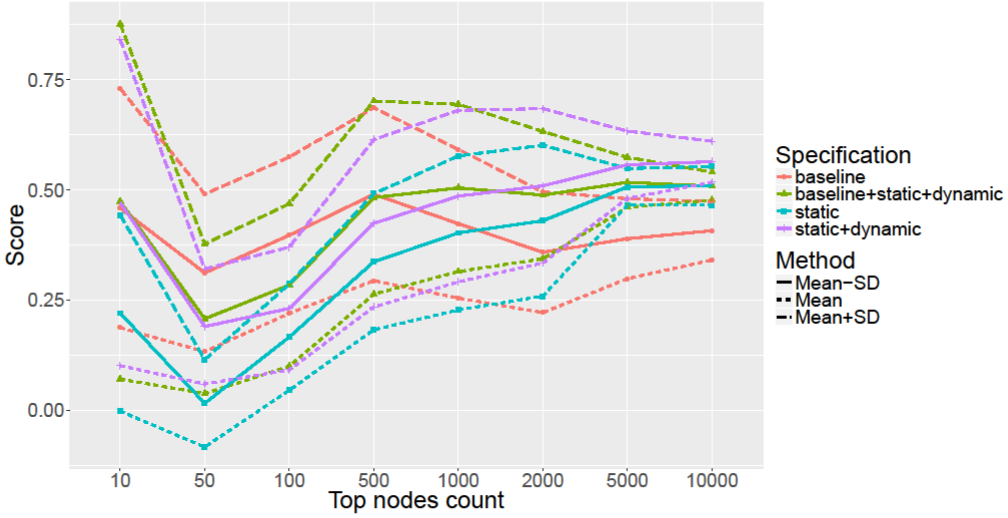


Figure 3.11 Rank predictions on Nepal set method stability—with methods GLM, RF and DL from top to bottom

For the prediction on the Sandy set, the Irene set is the default training set. GLM still provides the best results. The result on the Sandy 4.5 set is much better than the Sandy set. A possible explanation for this is that the diffusion has just broken out at the beginning time of Sandy 4.5, just like the case of Nepal set. This makes the predicting power of the features much better than ones from Sandy set, where the diffusion has not started.

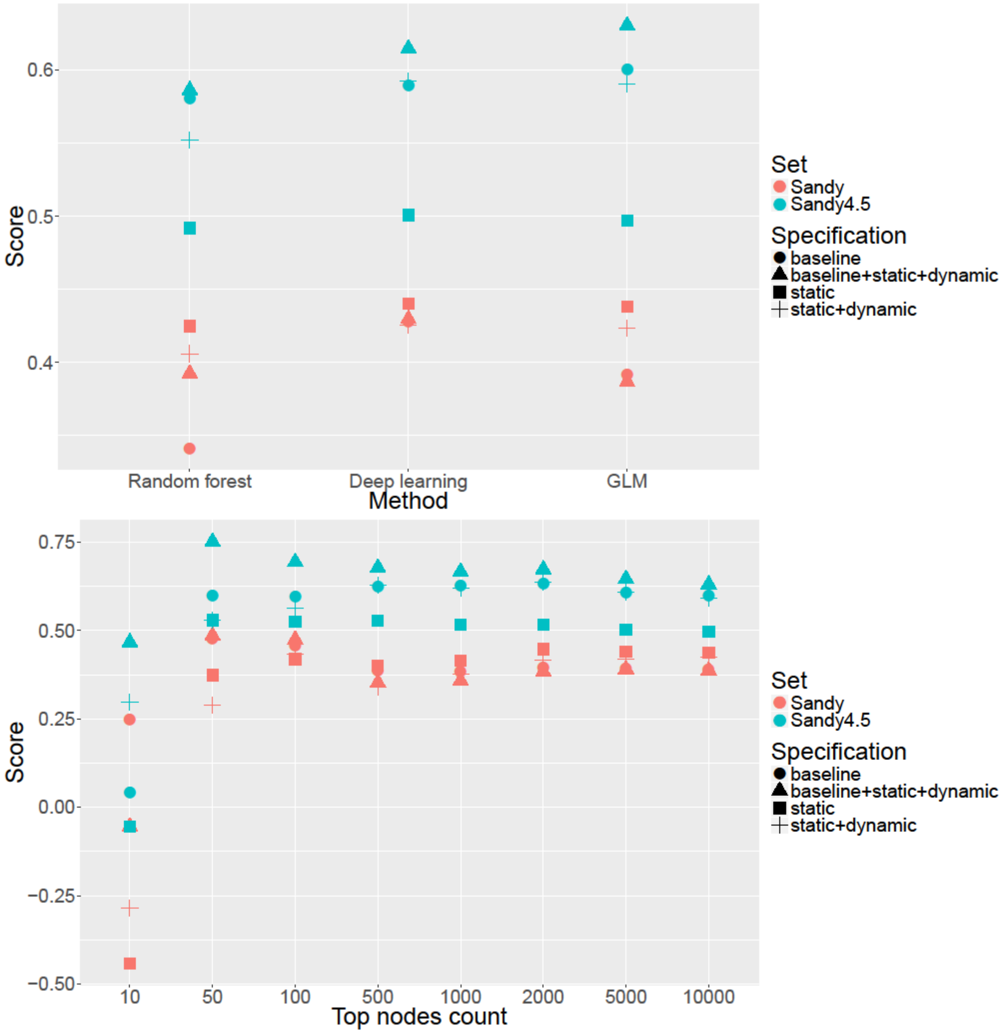


Figure 3.12 Rank predictions using GLM on Sandy set varying methods and testing sets

As to the prediction on Blizzard set, the training set is Irene+Sandy for the first methods comparison plot. Again GLM still gives the best on average results. Like in the Nepal case, different training sets do not give different results. The feature combination baseline+static+dynamic consistently produces the best results over various settings, which validates the fact that network statistics can indeed improve the prediction but the extent of improvement will largely depend on the testing set and other various settings.

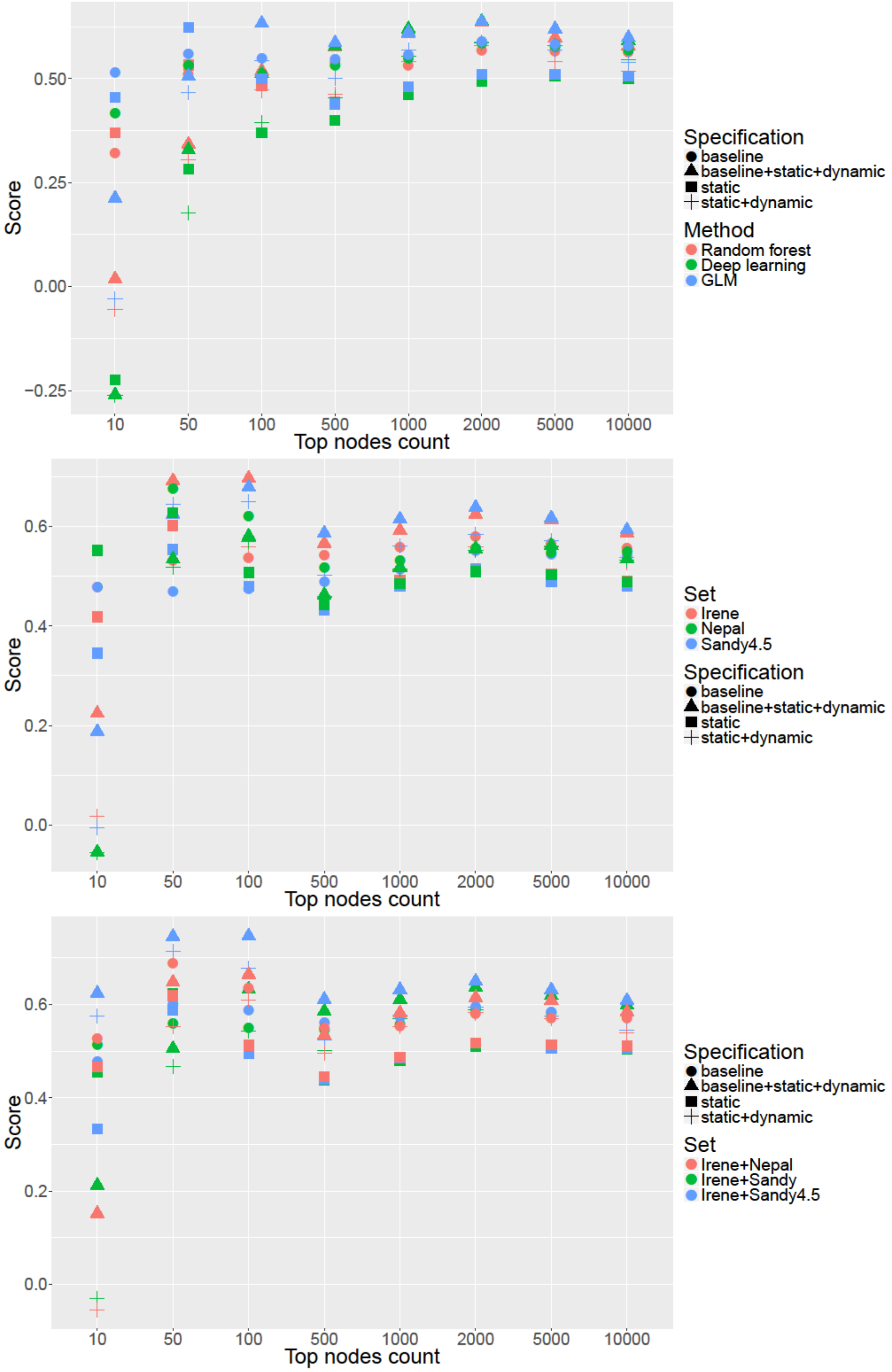


Figure 3.13 Rank predictions using GLM on Blizzard set varying methods and training sets

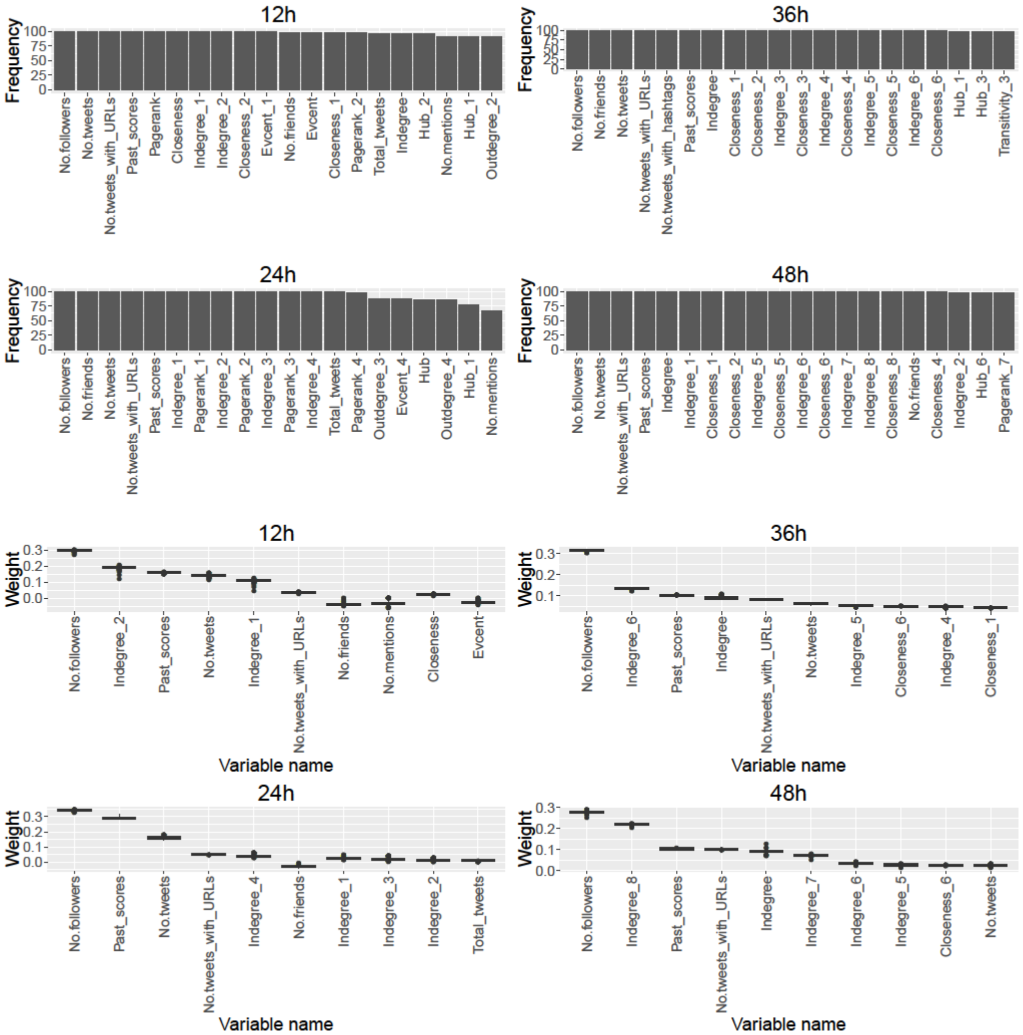


Figure 3.14 Rank prediction on Nepal set GLM variable importance

Figure 3.14 provides the variable importance measure for Nepal set rank prediction and the titles for each plot corresponds to various T values. The subscripts for the network features refer to the separate time windows of length Δ. All values are computed based on n=100 runs, where during each single run different features can be selected and different feature weights can be obtained. The upper histograms show the frequency of selection from the LASSO algorithm for features over 100 runs, and the lower bar charts show the distribution of feature weights for top 10 features ranked by the absolute values of their weights over 100 runs. Some baseline features—number of followers, number of tweets, number of tweets with URLs and past scores are always selected and exhibit high weights across different T values, indicating high predicting power consistently. But certain dynamic network features are also always chosen and have significant weights, sometimes even outperform the baseline ones. Moreover, I can see from the bar plots that all feature weights are robust over the 100 runs—again indicating the stability of the GLM method. In summary, the network features not only help the prediction, but also are also of great importance in the prediction—not add-ons features that just improve the results.

3.3.2 Classification

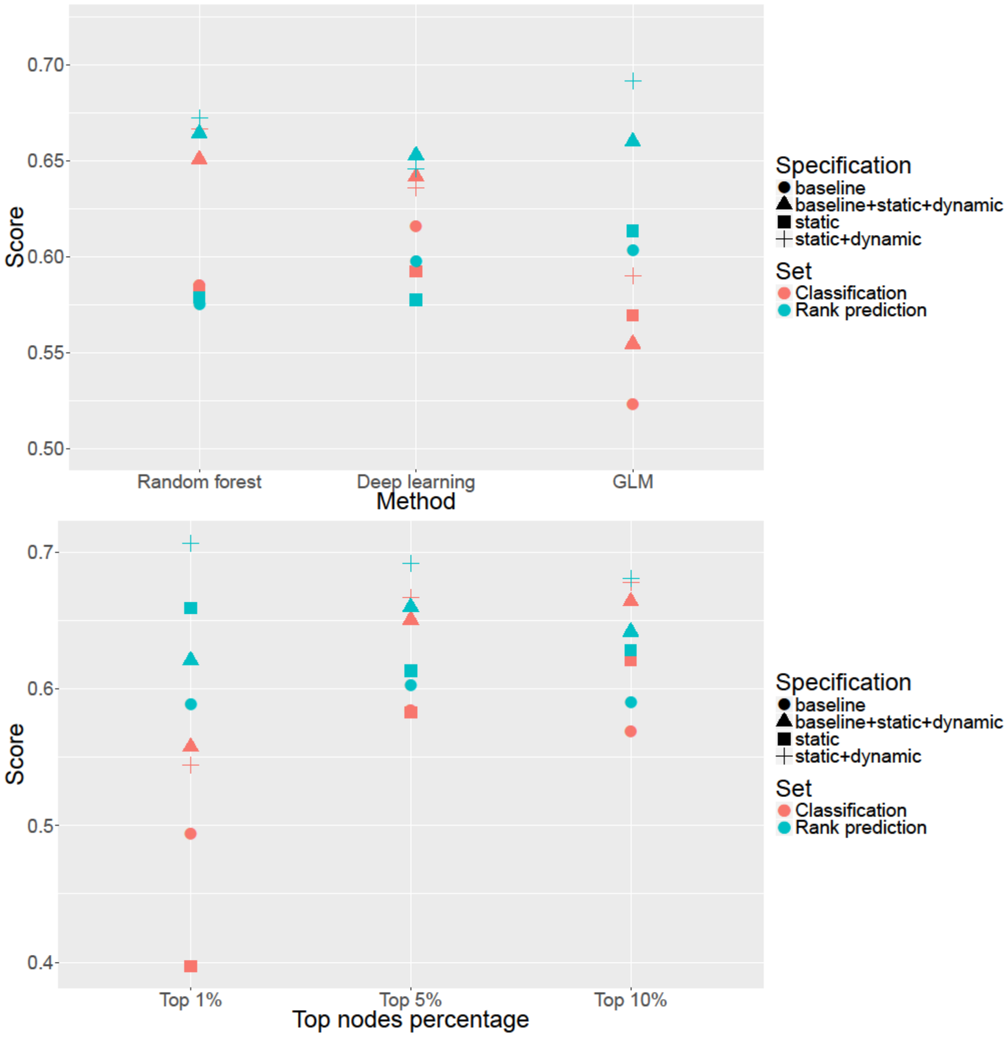
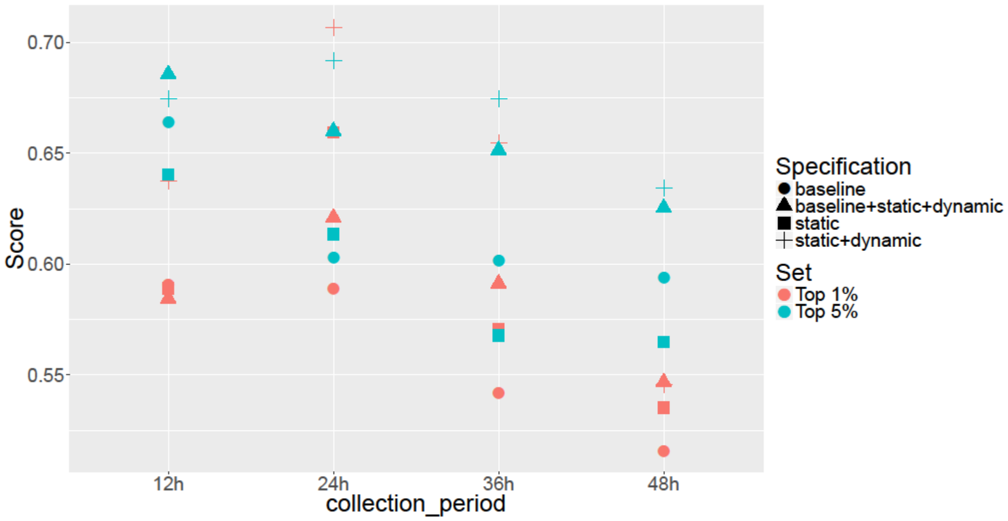
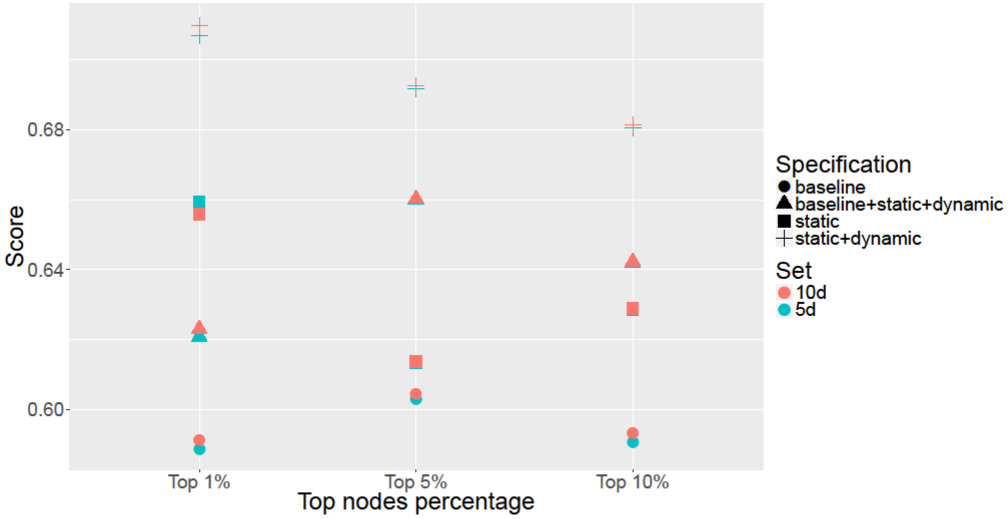


Figure 3.15 Top quantile users binary classification comparison for Nepal set—using classification and rank prediction

Figure 3.15 gives the results comparison for top quantile nodes binary classification using direct classification and rank prediction. The default method for rank prediction is GLM and the default quantile for prediction is 5%. The lower plot uses random forest for the classification since it is the best on average method for classification. It is interesting to find the rank prediction can also handle this task well, even better than the standard classification. Larger improvement is observed for method GLM, where the main reason may be the down-sampling in the classification since the quality of the CV operation in GLM depends heavily on the size of training set, and there is no training data loss due to down-sampling for rank prediction. Great improvement of rank prediction over classification with quantile 1% seems to support this hypothesis, consideration the sample size for 1% is only 400 for classification compared to 20k for rank prediction for each run.





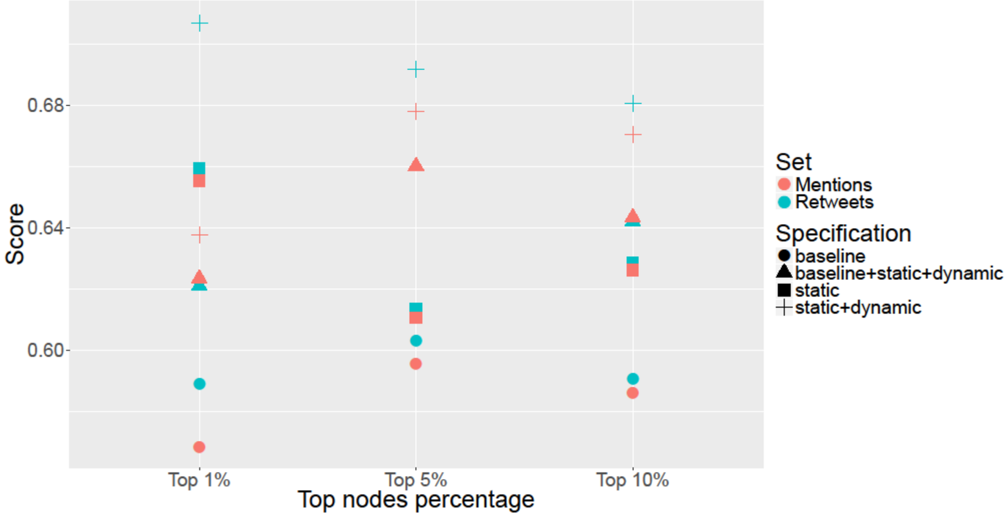
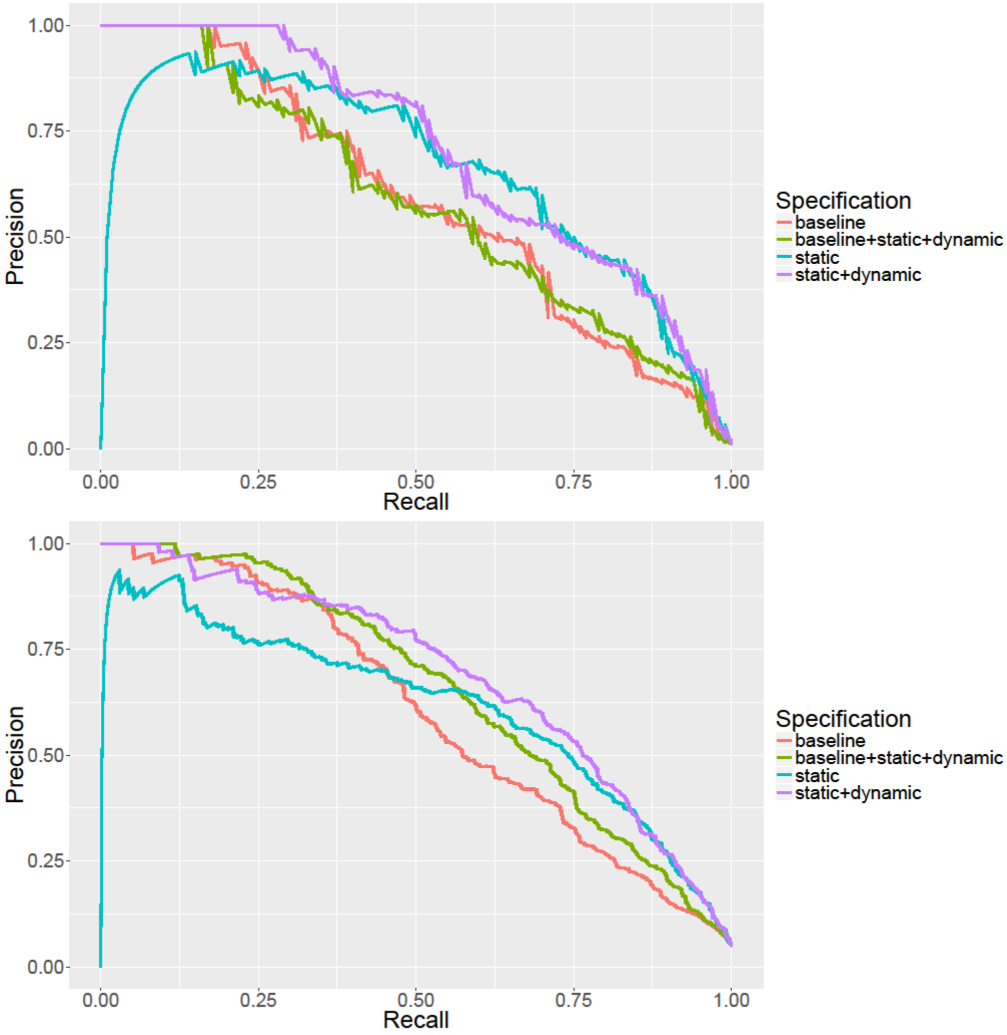


Figure 3.16 Top quantile users binary classification using GLM from rank prediction varying parameters

However, this down-sampling seems to be necessary for classification, the performance turns out even worse without it. The main reason that the rank prediction is well suitable for this task may simply be that it is a top rank nodes classification problem—meaning the positive class to be predicted is highly correlated with the rank. While for classification of user classes less correlated with rank, the results would be different. Figure 3.16 lists results from various settings for the top quantile nodes binary classification using rank prediction. They are similar to results from the ordinary rank prediction, which is natural. The order of the feature combination remains stable over various settings. Combined with similar results from rank prediction, I conclude the improvement of network statistics is robust.



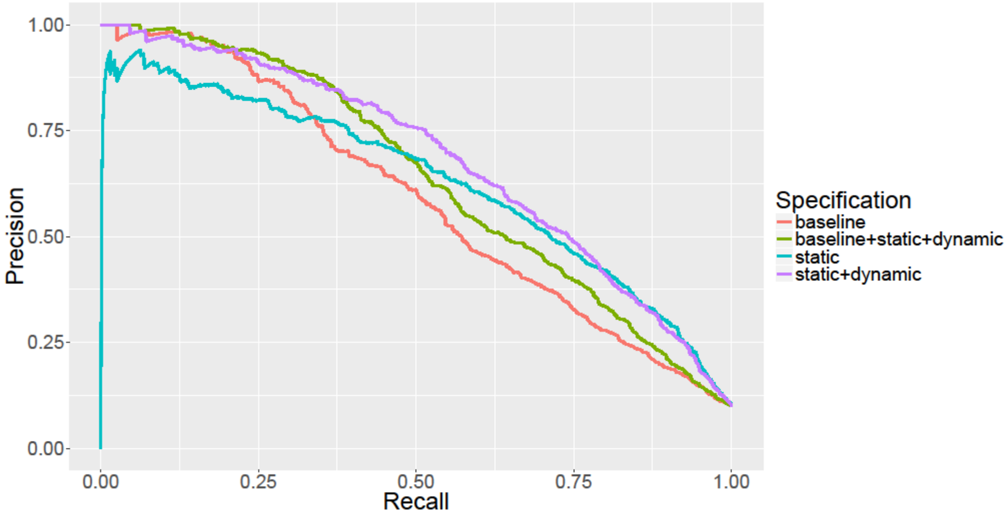


Figure 3.17 Top quantile users binary classification from rank prediction PR curves—1%, 5% and 10% from top to bottom

Figure 3.17 displays the actual precision-recall curves for three quantile values. The sharp rise for feature combination static is likely due to the first few top-ranked nodes by prediction scores do not include any true positive values. While the PRAUC value can be considered as an average measure of the performance, the PR curve gives the full dependency plot of precision and recall values. Again the improvement of network statistics is still robust across changing recall values.

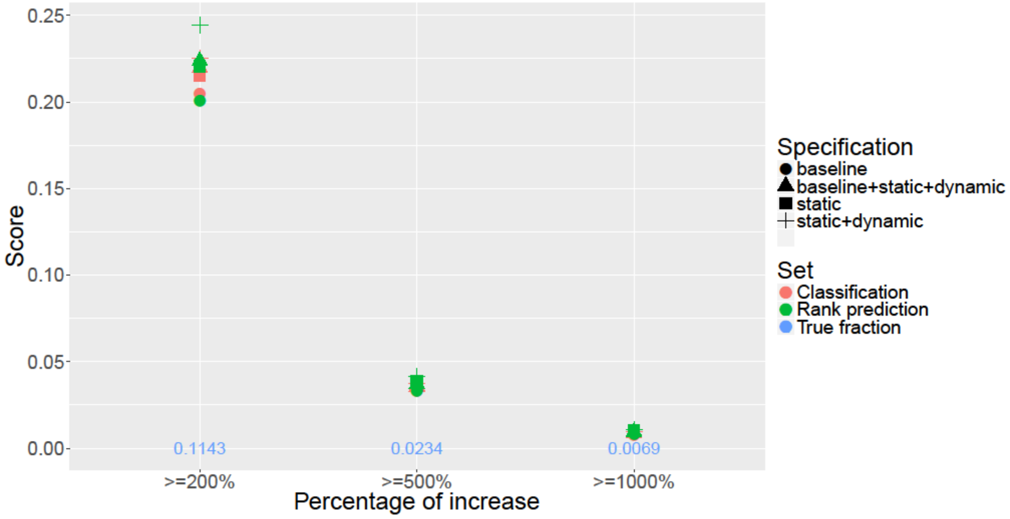


Figure 3.18 Rapidly increasing users binary classification from rank prediction PR curves—1%, 5% and 10% from top to bottom

Figure 3.18 shows the results for rapidly increasing nodes binary classification using standard classification (with RF) and rank prediction (with GLM). The x-axis stands for the percentage of increase, and y-axis uses the same PRAUC score as the top quantile nodes classification. The ordinary classification can simply be replaced by rank prediction since they are producing comparable results. Compared to the true fractions of these nodes, which can be considered as the approximate scores for the complete random model, the results are bad—only do twice better towards the random guess. The differences among various settings are not significant; this simply implies that my framework cannot handle this task well based on given data.

3.3.3 Direct prediction

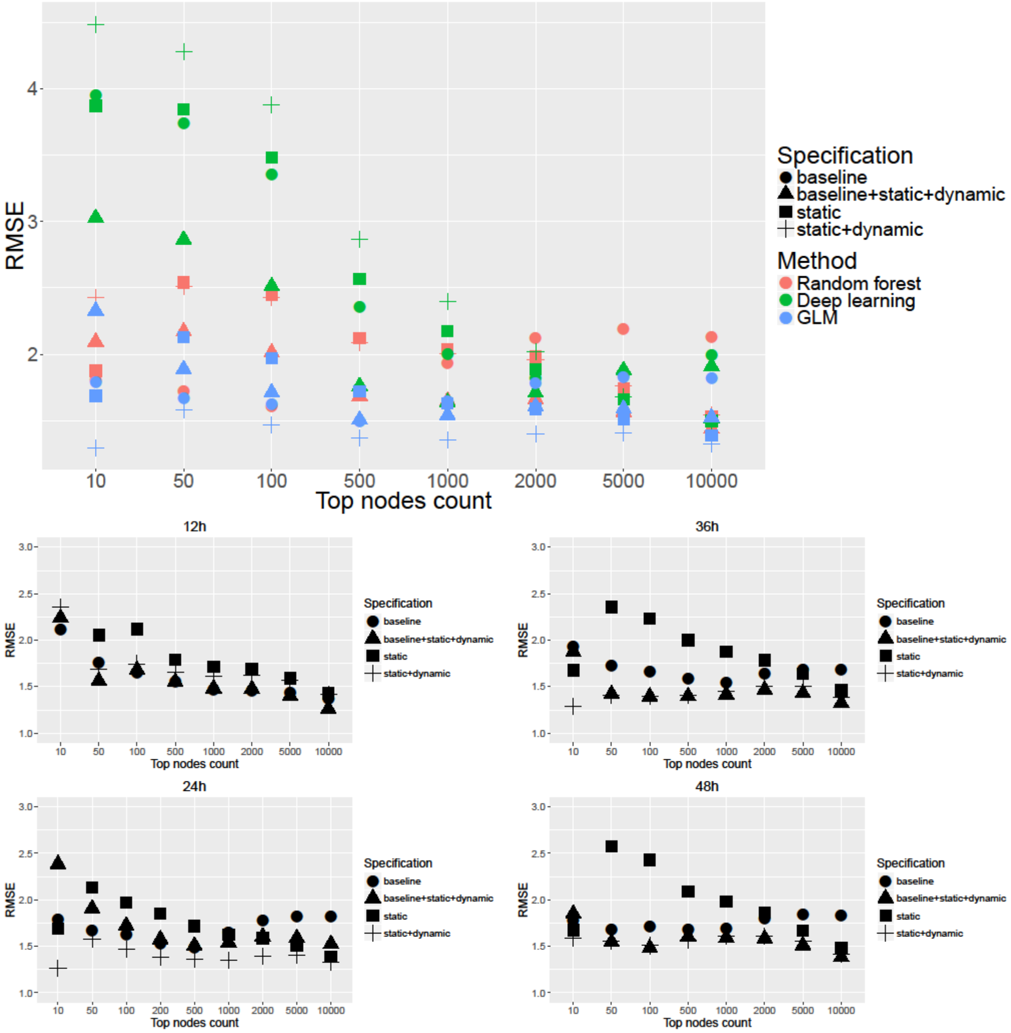


Figure 3.19 Direct predictions on Nepal set varying methods and T

Figure 3.19 gives the results of direct prediction on Nepal set varying methods and T, recall the default training set used here is Irene+Sandy. The GLM still performs the best, network features still improve the prediction consistently and dynamic network features still outperform the static ones. Considering the response distribution on the log scale, results with RMSE around 2 can be considered good. However, after I perform some extension experiments on direct prediction, I find the results are not as stable as ones obtained from rank prediction.

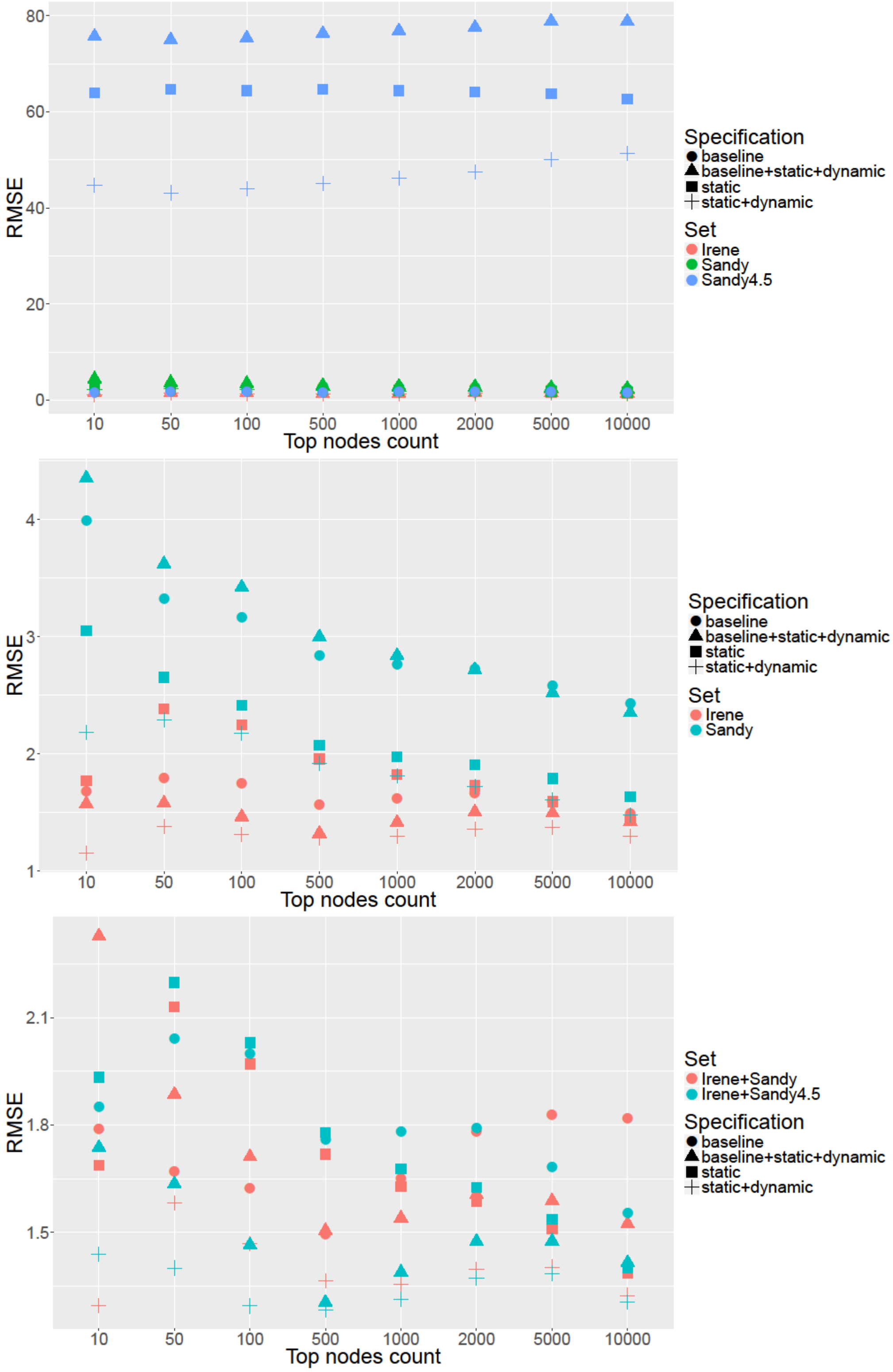


Figure 3.20 Direct prediction using GLM on Nepal set varying training sets

Set	Irene	Sandy	Sandy 4.5	Irene+Sandy	Irene+Sandy 4.5
Log past scores	0.394	0.740	0.742	0.561	0.277

Table 3.3 KS statistics of past scores between various training sets and Nepal set

Figure 3.20 lists the result of direct prediction on Nepal set by varying training sets. The second plot is the same as the first except results from Sandy 4.5 are removed for better visualization. Clearly the choice of training sets proves to be important for direct prediction, which is unlike the rank prediction case. The performance can go from pretty good to extremely bad if the training set is not properly chosen. The main reason lying behind should be the heterogeneity between the training and testing sets. However, there is no existing rigorous method for training set selection for my task here. Thus I try to propose some empirical criteria on my own. One potential candidate is using the Kolmogorov–Smirnov (KS) statistics, which is a commonly used metric to measure similarity between two probability distributions. Here I simply calculate the KS statistics of past scores between each given training set and the testing set. The KS scores are computed using the R base function "ks.test". The reason to use past scores is that I believe past scores represent past activity of users and should be most correlated to their future response. I can definitely compute the KS statistics for each feature that I think are important and aggregate the results to give a single output. However, given that the variable importance is not given beforehand, it is difficult to determine which features to choose and how to aggregate the resulting scores from each feature. Thus I will just use past scores to compute KS statistics for now being.

Comparing the prediction results in Figure 3.20 and KS statistics in Table 3.3, the KS statistics can indeed identify good training sets to some extent, though still far from perfect. Since the calculation of KS statistics depends on the sample size of the empirical distributions, I need to compare KS scores of single training set and double training sets separately. Following this, Table 3.3 tells Irene set is the best single set while Irene+Sandy 4.5 is the best double sets, which all fit the results well. However, the KS scores cannot tell whether Sandy or Sandy 4.5 is worse, thus my suggestion is just use the score to choose the best one (the one with significantly lower score) to use.

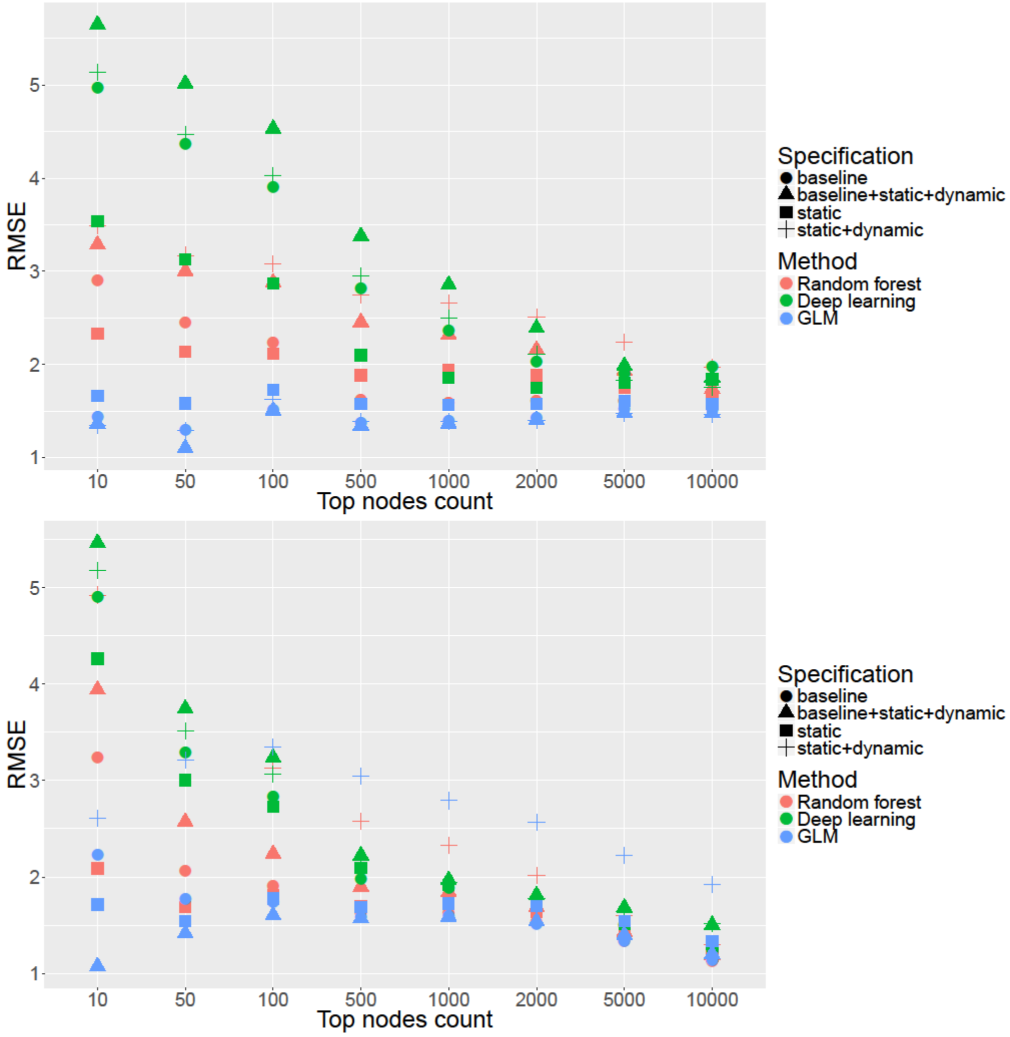


Figure 3.21 Direct predictions on Sandy 4.5 set and Blizzard set varying methods

Figure 3.21 gives the results for the direct prediction extension study on Sandy 4.5 set and Blizzard set varying methods. The reason to use Sandy 4.5 set simply is it returns better results for rank prediction than Sandy set. The leading position for GLM is still firm, though the order of feature combinations are not so stable compared to rank prediction.

Figure 3.22 shows the results on Blizzard set using GLM method with varying training sets. Similar problems as Nepal set prediction are observed, and it becomes even worse. Also the performance of network statistics suffer most from wrong choice of training set, although they still provide some improvement for the best training set. This is why the direct prediction is unstable.

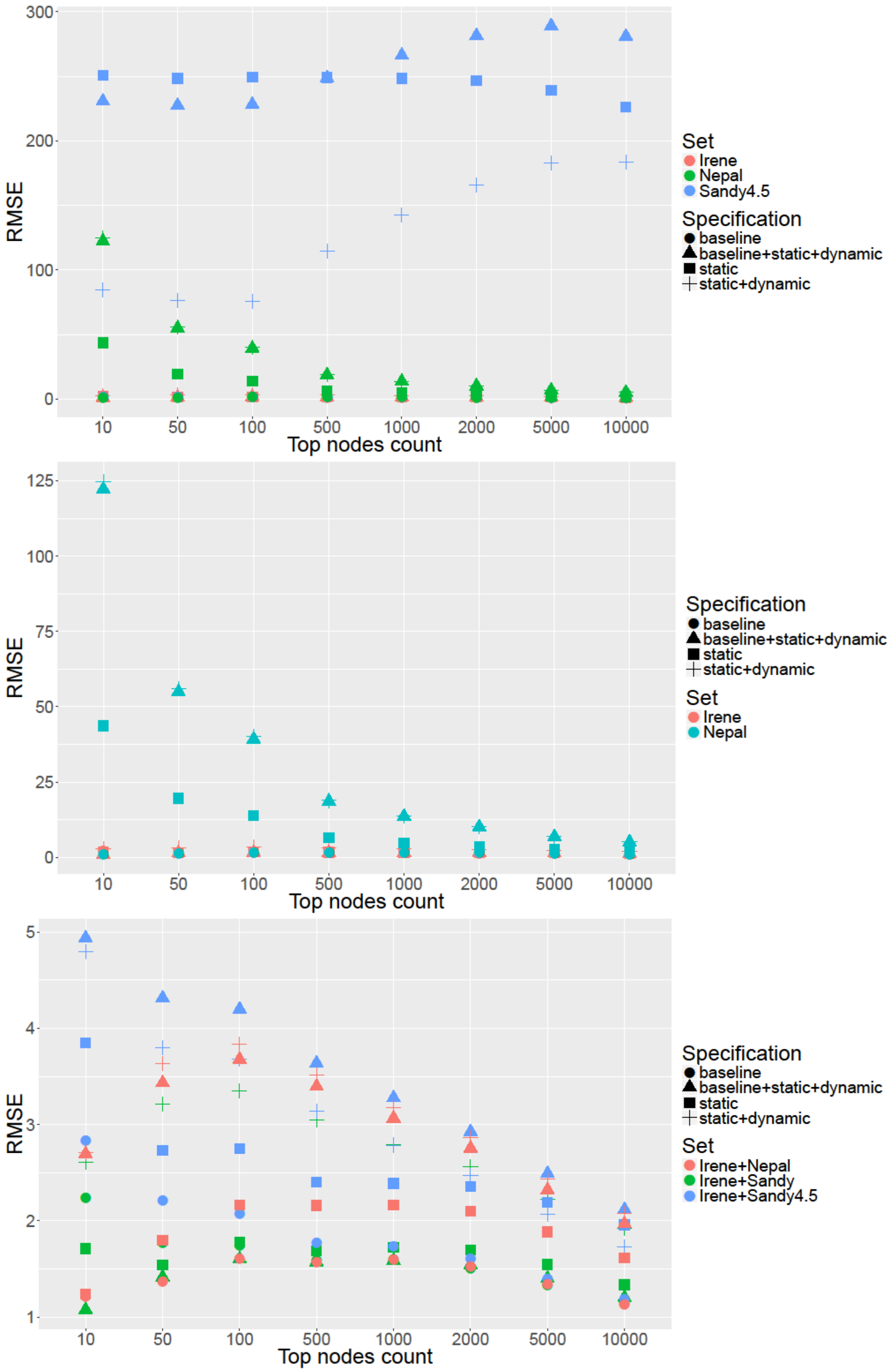


Figure 3.22 Direct prediction using GLM on Blizzard set varying training sets

Set	Irene	Sandy 4.5	Nepal	Irene+Sandy	Irene+Sandy 4.5	Irene+Nepal
Log past scores	0.380	0.900	0.590	0.102	0.444	0.444

Table 3.4 KS statistics of past scores between various training sets and Blizzard set

Once again, the KS scores provide good suggestion on choosing a proper training set. The best single set and double set given by KS scores is Irene and Irene+Sandy respectively, both of them are indeed the ones that give the best on average results. Since the network statistics could potentially suffer much degradation due to improper choice of training sets, I use the more stable baseline feature combination for the direct prediction task when I cannot pick a best training set from KS scores or other related metrics.

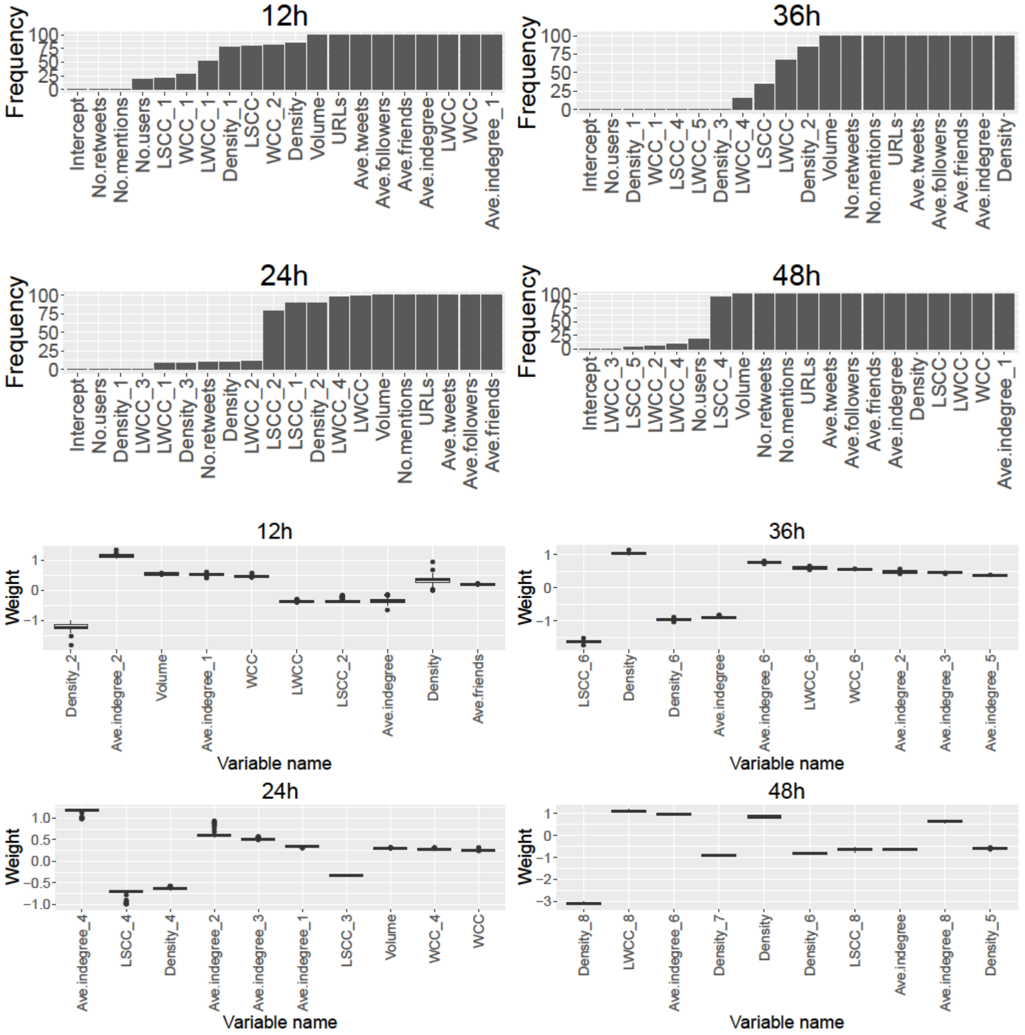


Figure 3.23 Direct prediction on Nepal set GLM variable importance

Figure 3.23 shows the variable importance of GLM method on direct prediction of Nepal set. The setting is exactly the same as the rank prediction case. Here I rank the frequency with increasing order in the upper selection frequency plot since most of the features are selected all the time. For the feature weights bar charts, there are more dynamic network features staying on this top10 rank than the rank prediction case, recall the features are ranked by the absolutely value of their weights. Therefore I believe the network statistics still have great potentials for the direct prediction task, but how well they can perform will heavily rely on a good choice of training set.

User_name (rank/log(score+1))	Past score	Predicted by SD	Predicted by B	Predicted by BSD	True
ActualidadRT	13/7.84	4/9.11	17/8.69	7/9.65	7/8.60
johnspatricc	**32/7.33**	**9/8.68**	**66/7.89**	**36/8.35**	**6/8.60**
AP	8/8.67	15/8.38	4/8.24	11/9.95	3/8.82
TurkKizilayi	238/5.83	181/6.26	516/4.90	301/5.93	2/8.83
ksushma140	**21/7.54**	**11/8.46**	**299/5.64**	**172/6.54**	**5/8.66**

Table 3.5 Prediction further investigation on selected nodes using GLM

In order to further investigate how the features are acting on various users, I selected some users as examples to see their individual prediction output. Table 3.5 shows the prediction results for 5 selected users. The short notation SD, B, and BSD refer to the feature combination static+dynamic, baseline and baseline+static+dynamic respectively. The values in the table are shown in the format of "rank/log(score+1)". For users "johnspatricc" and "ksushma140", the network features alone can make good prediction while the baseline features are off target. On the contrary, baseline features predict well on user "ksushma140" when the network features are off the target. And in both cases, combining baseline and network features will generate average outputs. As results from various settings show, baseline+static+dynamic is indeed the most stable feature combination and nearly always outperforms the baseline ones. However, for user "TurkKizilayi", none of the feature combinations can do a decent prediction. The main reason is that this user is simply not active in the past if I look at the past scores, thus I cannot obtain informative features. Thus this brings one drawback of the framework—it does not well on users with poor activity, which is something hard to avoid for this type of urgent diffusion prediction unless additional information can be obtained from external resources.

Thinking on the side of stake holders who want to find potential influential users, the actual scores or rank of the users might not be a major concern, where the goal is to find a list of potential candidates. In that case, the main goal is to discover how many "real" top nodes are missed from each feature combination. Table 3.6 shows the missing count by the three feature combinations plus the missing count by the union of top nodes predicted by all three feature sets.

Missing counts by rank	Missed by SD	Missed by B	Missed by BSD	Missed by all	Count of the union
Top 10	5	6	7	4	16
Top 20	11	12	11	8	33
Top 50	19	30	24	15	75
Top 100	40	45	44	27	144

Table 3.6 Missing nodes count under various feature sets using GLM

For instance, in the first row it shows among the real top 10 nodes ranked by the true response, only 5 are captured by top 10 nodes given by feature set static+dynamic and only 4 are captured by top 10 nodes given by feature set baseline etc.; if I union the top 10 nodes from three feature combinations together, I can capture 6 of real top 10 nodes but the total number of nodes I am using will increase to 16. Here I illustrate an alternative way to solve the problem, while whether using the union to achieve the maximum coverage is subject to the practical implementation considerations—e.g., what is the cost of targeting each user and what is the benefit if I correctly specify one more user.

3.4 Discussion

3.4.1 Summary

In this Chapter I have described a comprehensive study that is across various settings on user influence prediction during urgent event diffusion. The main findings are:

1. The network features as defined, especially the dynamic network features can help the prediction and improve the prediction performance over baseline features across various settings consistently. The dynamic features outperform the static ones consistently—implying the additional dynamic information is helpful in the prediction. Similar results are shown by both the prediction results and the variable importance ranking. The extent of improvement largely depends on the testing set and other parameters.

2. GLM proves to be the best prediction method considering various factors: prediction performance, interpretability, variable importance and runtime efficiency etc. The good performance of GLM implies that there could be strong log-linear dependency among the selected features and the response. In contrast to the

conventional impression that the nonlinear models will usually outperform the linear model, the two nonlinear methods involved in my prediction task perform similarly or even slightly worse than the linear method. One direct reason for this could be that there is no significant nonlinear relationship between the selected features and the response.

3. The rank prediction demonstrates the most stable and reliable result, while the result for direct prediction is not stable and highly depends on the choice of training sets and testing sets. Moreover, it is shown that both classification tasks—top quantile nodes and rapidly increasing nodes classification can be solved well using rank prediction solely.

4. KS statistics can potentially serve as an empirical criterion to choose training set for direct prediction, which proves to be of extreme importance.

5. Both baseline and network features do not predict well on nodes with poor activity, nor for the rapidly increasing nodes classification task. The main reason behind is similar—the features used are not informative, external resources or data is needed to handle this problem.

In summary, I have established a novel early time prediction framework that utilizes all of the limited information that would be available in an urgent diffusion context and still achieves reasonable results. Moreover I have shown that the newly adopted features from the underlying diffusion network have consistently impressive predicting power on the future user influence.

3.4.2 Future work

One crucial future direction would be to extend this framework to other types of urgent events other than the natural disasters used in my work, which should output comparable results, at least in the Twitter context. Another promising direction is to evolve the whole framework would be to other platforms—e.g. Facebook or Instagram, which again should not be difficult to do since these social media share much in common and as noted in the introduction part—many models and methods can be applied interchangeably among these platforms. Finally, provided with enough time and engineering effort, it would be possible to build a user friendly API to implement the framework that would interface with the Twitter API to provide real-time online prediction for any user of the API.

Chapter 4: Predict topic popularity during urgent event diffusion

4.1 Introduction

4.1.2 Motivation

In the previous section, I discussed two well—cited studies on hashtag popularity prediction and the main reasons why they are not well fit for my prediction task. In this Chapter, for the hashtag prediction task I describe the use of the number of retweets a hashtag can obtain in the prediction period as the prediction goal. Generally speaking, the framework for hashtag popularity prediction is similar to the user prediction case since they both belong to the urgent diffusion prediction category. Since the behavior of hashtags can be somehow considered as an aggregation of many users' behavior, there should be something more general in common between the two tasks. Thus I propose to make the general settings of the two tasks as similar as possible. I also adopt two sets of features—baseline and network features for the hashtag prediction, corresponding to ones used in the user case. Moreover, since I do not adopt any text content features for the user prediction, I also remove them from the general settings for hashtag prediction. But to compare with previous studies, I also show how the results can vary if all available text content features are included.

As the topic prediction part, I will use the LDA model for topic modeling and propose prediction on generated latent topics. One important question left to determine is the definition of documents. Godin (Godin, 2013) uses single tweets as documents to train LDA models for hashtags recommendation towards general tweets not including hashtags. But Hong (Hong L. a., 2010) has pointed out that the effectiveness of topic models can be highly influenced by the length of the "documents", and for better topic modeling aggregation of short messages is recommended. Ma (Ma, 2013) considers hashtags as documents and uses topic probabilities derived from LDA model as text content features for hashtag popularity prediction, suggesting hashtags are potential candidates for LDA documents. Moreover Mehrotra (Mehrotra, 2013) points out pooling of tweets by hashtags can provide an improvement over LDA topic modeling of single tweets. Therefore in this task I decide to simply treat hashtags as documents and perform LDA topic modeling only on tweets containing hashtags, this is due to the similar reason I mentioned earlier in literature review part: I do not have the time and interest on specifying related hashtags for general tweets since it is not the main focus of my study. Further details on hashtags text cleaning and LDA modeling are mentioned in the next section.

In this Chapter I will show there are a great similarity yet some slight differences between prediction of user influence and prediction of hashtag popularity. In both tasks I find addition of network statistics can have significant improvement on the prediction performance over baseline features, also the linear method outperforms the nonlinear ones consistently—possibly indicating strong log-linear existing between selected features and target response for both prediction tasks. The LDA topic prediction shows a slight different pattern since each topic is a weighted aggregation of all the involved hashtags.

4.1.2 LDA model implementation

Before implementing the LDA model, I need to preprocess the text first. The steps for the text preprocessing are exactly the same as in Chapter 2 for hashtags clustering and classification analysis (see Figure 2.7). One extra step to do before the text cleaning is filtering. Here I adopt a simple filtering strategy: removing hashtags with unknown digits or including only one digit, by doing so many noisy and irrelevant hashtags will be removed. More involved and rigorous filtering may include spam hashtags detection (Stringhini, 2010) (Castillo, 2011) or hashtags semantic clustering (Costa, 2013) (Ozdikis, 2012) (Vicient, 2014). However, these hashtags filtering and cleaning approaches often require well labeled data to train the models and validate the results, which are not available to me; moreover, my main goal is to explore the early time prediction framework for urgent diffusion on Twitter, and I believe hashtags filtering will not affect the performance much but will indeed matter for practical implementation where the quality of the underlying topics will be a big concern.

After the simple filtering is performed, each hashtag is considered as a separate document and the text of the document is simply the aggregation of all tweets including that hashtag. The text is processed following the steps showing the Figure 2.7. Then the top 10k tokens, across all documents, will be used with tokens occurring over 50% across all documents eliminated to remain more distinguishable tokens among the documents. Finally TF-IDF weighting is applied to improve the modeling performance.

Generally speaking, the Latent Dirichlet Allocation (LDA) model, is a flexible topic model to extract latent topics from a collection of documents while each document is considered as a mixture of various topics with the probabilities provided by the model. This is where the flexibility comes from—each document can either be more concentrated on fewer topics or be more diverse on more topics. Another advantage of LDA is the generalizability to new documents, compared to the other popular topic model— probabilistic latent semantic analysis (pLSA), of which the size of parameters is increasing linearly with the number of training documents and causes serious overfitting problems.

Figure 4.1 shows the plate notation for the ordinary LDA model. Here I assume there are M documents, K underlying topics and each document has the same number of N

words for simplicity. Since the general framework is Bayesian, each parameter involved is a random variable. Among these variables only the word identification variable—w is known to me; thus it is in grey color and rest unknown variables are in white. More specifically, the meaning of each variable is:
α: the Dirichlet prior parameter for document-topic distribution
θ: the multinomial topic distribution probabilities varying from each document
z: the topic assignment for each word within each document
β: the Dirichlet prior parameter for topic-vocabulary distribution
φ: the multinomial vocabulary distribution probabilities varying from each topic
w: the identification for each word within each document

The model considers each document as a collection of words, and the following is the general data generating process:
1. Generate the document-topic distribution θ from the prior α for each document
2. Generate the topic-vocabulary distribution φ from the prior β for each topic
3. For each word within each document, draw the topic assignment z from the distribution given by parameter θ, then draw the word w from the topic-vocabulary distribution φ with the topic assigned as z
This data generating process explains the high flexibility and generalizability of the model, and it can be further adapted to different situations by various extensions and derivations.

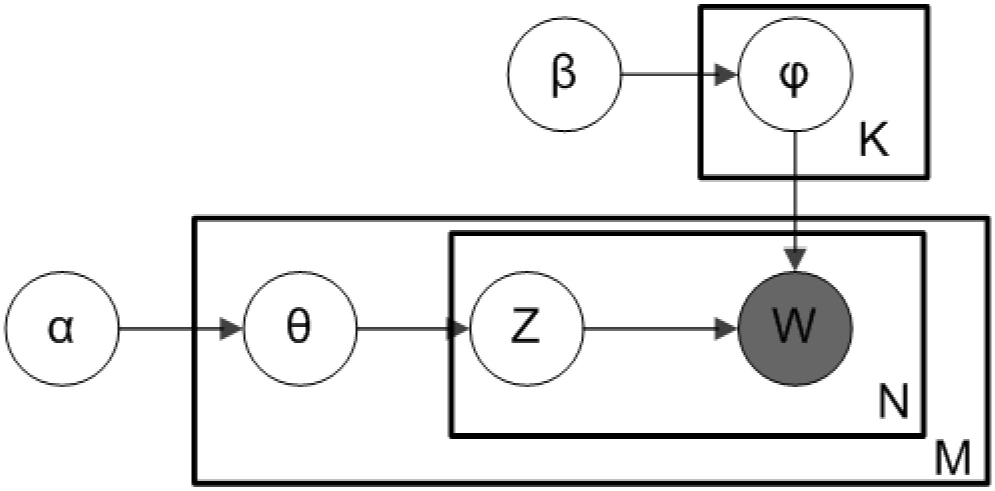


Figure 4.1 LDA model illustrations
(source: https://en.wikipedia.org/wiki/File:Smoothed_LDA.png, this file is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license)

As to the model inference, since the target posterior distribution is intractable, approximate approaches are implemented. There are two state-of-art inference approaches for LDA model—collapsed Gibbs sampling (Porteous, 2008) and variational Bayes method (Blei, 2003). While the collapsed Gibbs sampling integrates out unimportant variables to improve convergence speed, variational method chooses a simplified distribution to approximate the target by minimizing the KL divergence. In my study, I implement the LDA model using the python library genism (see https://radimrehurek.com/gensim/ for more implementation details), which adopts an online version of variational Bayes method. The algorithm is better scalable to larger collection of documents and can update the model continuously. For my task, estimates of the document-topic probabilities are the only ones that I care about. In most results I will adopt the default settings unless otherwise specified.

An important parameter to be fixed before I apply LDA modeling is the number of topics for the model. Ideally this value should be determined by some external evidence—human judgement, extra information about the documents or some supervised leaning tasks associated with the underlying topics etc. If no external resources are available, like in my case, I have to resort to some internal metrics. One of the intrinsic metrics often used for topic quality measurement is the topic coherent score. Topic coherent scores are generated to measure content consistency and coherence within each topic. There are various definitions for the coherent scores, in my study, I will just stick to one of them—the UMass score (Mimno, 2011). This score calculates the conditional probabilities of less frequent terms on more frequent terms where all terms are from top ranked words by frequency from each topic. Recalling that I have adopted the TF-IDF weighting before implementing the LDA model, thus the TF-IDF weighted version of corresponding probabilities are used for both model inference and coherent scores calculation.

Number	5	8	10	15	20
Mean	-341	-354	-364	-377	-387
SD	22	15	21	14	11

Table 4.1 Average topic coherent scores distribution varying number of topics with 100 repetitions for the whole Nepal set

Table 4.1 shows the mean and standard deviation of average coherence scores across topics after n=100 repetitions under various number of topics for the whole Nepal set. More specifically, under a given number of topics, top ranked (top 20) terms from each topic of the whole Nepal set are selected to compute a score and then the average score from all topics is obtained; the process is repeated by 100 times, the corresponding mean and standard deviation of the mean scores are listed in the table. Simply judging by the mean value of the average score, the coherent score increases as the number of topics decreases. However, the standard deviation is large so that it can just fill the differences, implying the results are unstable. One possible explanation for this unstability could just be that the topics derived from the model are not table. Moreover, this is only a relatively static calculation, the actual number for the topics may change over time and the content of the documents (hashtags) can also drift. Since the main goal of my task is to check how the prediction will vary between the appearing topics—hashtags and the abstract topics, I believe the actual quality of the topic modeling is of less concern. Out of this consideration, I simply choose an intermediate number 10 as the number of topics for my task and I believe the obtained conclusion is well generalizable to other cases with different number of topic.

4.2 Prediction on hashtag popularity

4.2.1 Response distribution

For the hashtag prediction task in this section, the setting and the framework are similar to the user case. So I simply repeat the same analysis done in the previous Chapter to see how these two prediction tasks will vary from each other. The same sets of data are used except for Irene set, which is excluded since it does not have many hashtags involved. Figure 4.2 shows the response distribution of various data sets under the default setting: T=1d, h=5d and k=10k. However, I notice that for Nepal and Blizzard sets the total number of hashtags that has any retweets within period T is only about 7000-8000, thus the actual number of hashtags involved in prediction is not 10k here. The distribution patterns are similar to what I have observed for the user case: power law distribution and comparable normalized response distribution. Figure 4.3 shows the percentile and ratio of increase distribution for hashtags from Nepal set, the distribution is more extreme (the differences among the points are much larger) compared to the user case: the red threshold line raises about 20% in the quantile value (with the same threshold value=10) and the magnitude increases by about 2 for top quantile points.

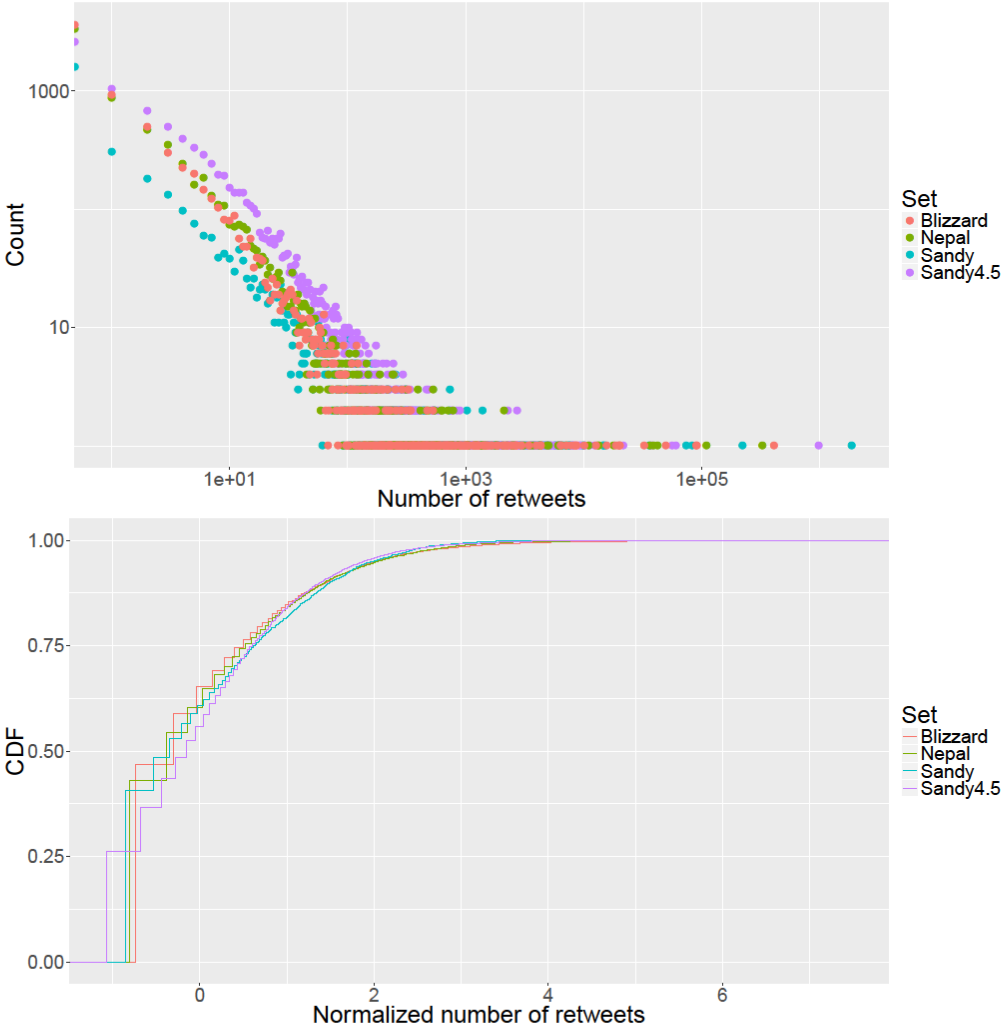


Figure 4.2 Hashtag response distribution with various data sets with default parameters

This "the rich get richer" is similar to the famous "Matthew effect" (Merton, 1968). The differences between the distribution of user score and hashtag score demonstrates this Matthew effect is much stronger for hashtags than users, which I believe is a natural result from the fact that the number of hashtags is much less than the number of users involved.

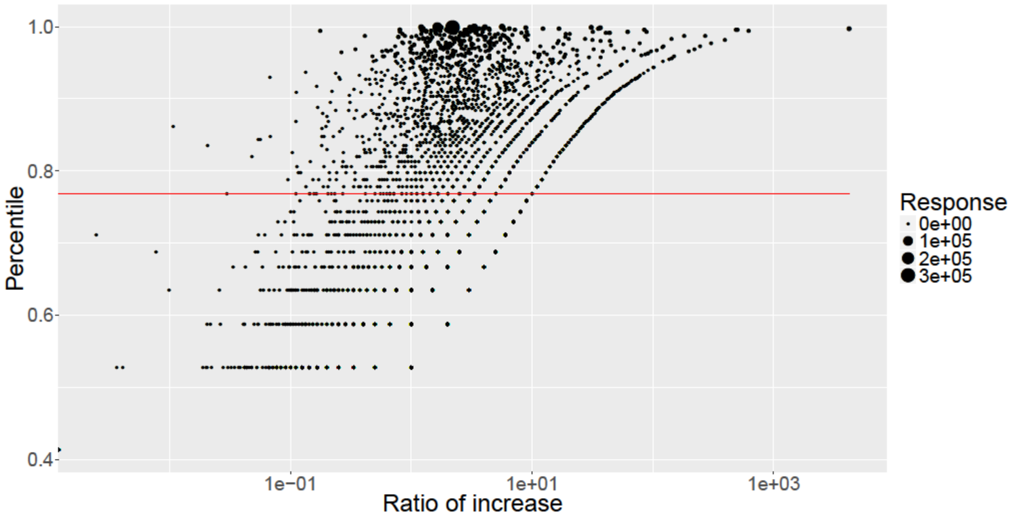


Figure 4.3 Hashtag scores percentile and ratio of increase distribution for Nepal set with default parameters

Table 4.2 is the response distribution summary from various data sets with varying T. Here the Sandy set refers to the Sandy 4.5 set and I will use this reference for all prediction tasks in this Chapter.

		1d	2d	3d	4d	5d
Sandy	**Total**	62755	80394	87838	93743	99759
	75%	1	0	0	0	0
	95%	18	9	6	4	2
	>=10	4793(0.729)	3828(0.728)	3440(0.744)	2816(0.783)	2030(0.833)
	>=100	879(0.914)	692(0.899)	607(0.904)	469(0.932)	299(0.960)
Nepal	**Total**	7800	11861	14457	16573	18309
	75%	8	3	2	1	1
	95%	110	51	35	25	19
	>=10	1795(1.000)	1739(0.969)	1670(0.947)	1575(0.928)	1460(0.927)
	>=100	424(1.000)	368(0.986)	334(0.991)	306(0.987)	267(0.978)
Blizzard	**Total**	7812	16018	19267	20991	21943
	75%	6	1	0	0	0
	95%	75	15	7	4	2
	>=10	1524(1.000)	1145(0.948)	719(0.940)	484(0.942)	308(0.954)
	>=100	312(1.000)	166(0.964)	97(0.969)	64(0.984)	38(1.000)

Table 4.2 Hashtag response distribution over varying T among various data sets (h=5d)

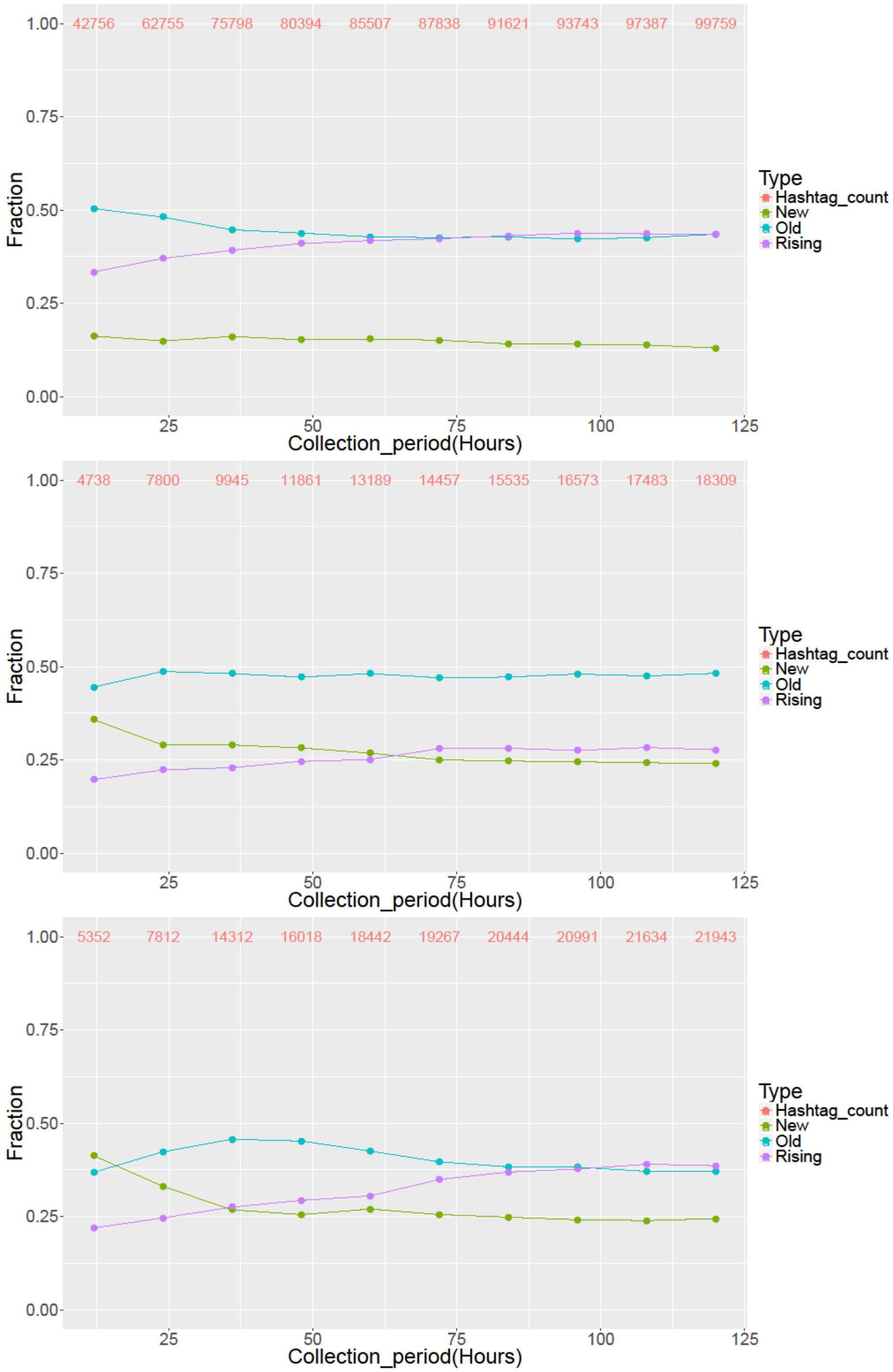


Figure 4.4 Top 1k hashtag components analysis over varying T values (h=5d) for Sandy set, Nepal set and Blizzard set from top to bottom respectively

All the specifications of Table 4.2 follow exactly the same meanings as Table 3.2. As stated before, the total number of hashtags is much less than the number of users in the same settings, especially for Nepal set and Blizzard set. This makes the distribution more concentrated: top quantile values are larger and there are larger fractions of "active" hashtags (hashtags with response greater than 10 and 100) within top 10k hashtags (recalling the numbers in parentheses are the corresponding fractions within the top 10k hashtags ranked by the response). Thus top 10k hashtags better represent the population of "active" hashtags than the user case. Also the number of "active" hashtags is dropping over time: it implies the gradual decaying of the diffusion and coincides with the observation for the user case.

Similar to Figure 3.7, Figure 4.4 shows the components evolution of the hashtags over time. But since I have much less number of total hashtags, I choose a smaller threshold for hashtags components observation—top 1k hashtags as opposed to top 10k users for the user case. Here I follow the same definitions for the three types: old ones refer to hashtags staying in the top 1k rank within past period (period [0,T]), rising ones refer to hashtags existing in the past and staying below the top 1k rank, and new ones refer to hashtags not existing in the past. All fractions will correspond to the fractions over top 1k hashtags by response during period [T,T+h]. Recall the Sandy set here is actually the Sandy 4.5 set, thus it is not surprising to see the fractions for all three types of hashtags remain steady over time since the diffusion turn to be steady during that period for Sandy. For the Nepal and Blizzard set, the fraction of new hashtags follows a gradually decreasing trend which means the diffusion process has turned from eruption state to steady state. While the old hashtags takes the lead at most times for all three sets—showing relative higher stability in the ranking compared to the user case, the behavior of rising ones is slightly different—showing differences of hashtag dynamics among three data sets.

To summarize, compared to the user case, similar yet more concentrated distributions exist for hashtags, and relatively consistent dynamic patterns from evolutions of both response and node components persist. All these similarities seem to demonstrate the same underlying truth about the urgent diffusion regardless of the diffusion media (users or hashtags) I have chosen. Thus consistent prediction results should also be obtained for the hashtag case and I will show it in next sections.

4.2.2 Experiment factors

The basic experiment settings will be the same as the user case. I will set up a similar series of experiments compare these two prediction tasks on every aspect. Thus I will only mention the parts that will be different. Since now the prediction is based on hashtags, the extracted features will be different from the user ones. I am still going to have two general types of hashtag features: baseline features and network features. Apart from these two general types, there are also other features for extensive studies.

The general baseline features will include:

'Number of tweets', 'Number of retweets', 'Number of mentions', 'Number of tweets with URLs', 'Number of users', 'Average user tweets', 'Average user followers' and 'Average user friends'.
All the above features are derived from tweets containing specific hashtags.
And the general network features (both the static and dynamic ones) include:
'Average indegree', 'Density', 'LSCC', 'LWCC' and 'WCC'
As opposed to the node level features for the user prediction, these network features are global level ones derived from the retweet networks formed from each hashtag.

Other additional features include network features from the co-occurrence (COO) network and text content features from hashtag tweets text. The COO network is defined to capture the co-occurrence relationship among hashtags. For tweets including multiple hashtags, I form pairwise undirected edges between all hashtags showing up in the same tweets. When users put multiple hashtags in the same tweets, it is actually indicating those hashtags are somehow related from other, at least through the content of the tweets. Therefore the motivation to setup this network is to characterize the relationship among various hashtags. I would like to know if features derived from this network will have any impact on predicting the hashtag popularity.
The network features with both static and dynamic versions include:
'Degree', 'Pagerank', 'Eigenvector centrality', 'Closeness centrality', 'Authority score', 'Hub score' and 'Local transitivity'.
All these are node level features defined on the undirected COO network. The text content features are derived from the tweets text—including the hashtag text itself. These features are derived following recommendations from the previous work (Tsur, 2012) (Ma, 2013) and availability to me, which contain the following:
'Hashtag character length', 'Hashtag digits indicator', 'Number of tokens', 'Average polarity', 'Clarity score'. The first two are simply features from the hashtag text. The other four are features from the text of hashtags. The polarity score is computed to measure how extreme a given tweet is. The score ranges from -1 to 1: -1 for extremely negative content, 1 for extremely positive content and 0 for neutral content. Average polarity is the average polarity score for each tweet from a given hashtag. Since I am doing log(x+1) transformation, I add 1 to all polarity scores to avoid getting invalid values. Clarity score is calculated in the same way as (Ma, 2013), which is the KL divergence between the token distribution within hashtags and the distribution within text of all hashtag tweets. This score serves as a quantitative measure to detect the distinction between document text and the background text—thus the "clarity" of the document. These additional features are adopted for an extension study to identify whether they can improve the prediction in addition to the above two general types.

4.2.3 Rank prediction results

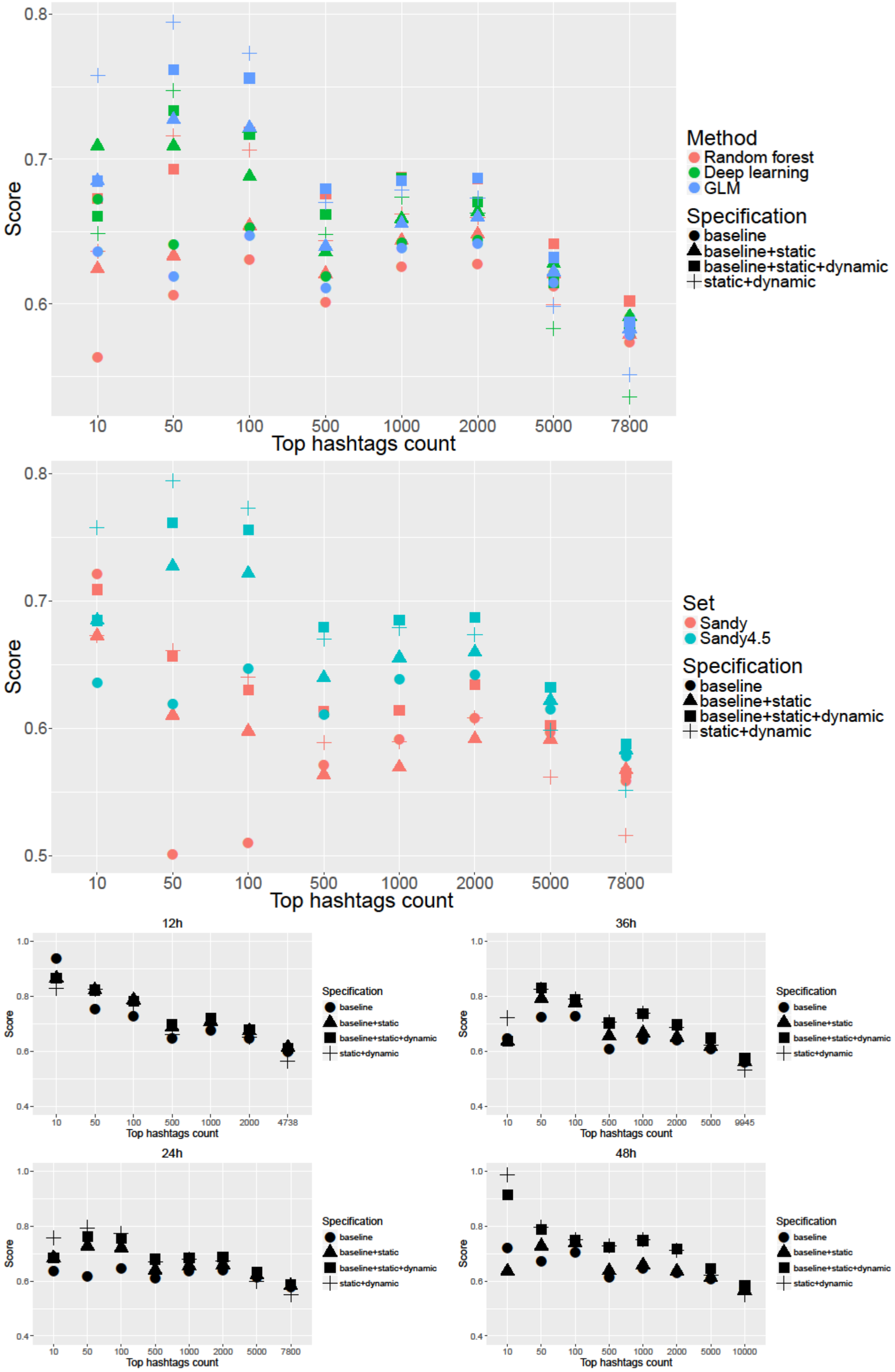


Figure 4.5 Nepal hashtag rank prediction results varying by methods, training sets and T

Figure 4.5 shows the results for Nepal hashtag rank prediction varying by methods, training sets and T, with exactly the same setting as the user case for direct comparison. Default values are assumed for prediction parameters not mentioned explicitly. All the patterns observed in the figure are similar to the user case:
1. GLM still does the best overall job and the differences among methods are subtle when number of nodes involved for the rank prediction is large.
2. The retweet network features improve the prediction when added to baseline ones.
3. Dynamic features not only do steadily better than static ones, but also outperform baseline ones for themselves alone when the number of nodes involved is not large.
4. Consistent results are observed by varying T values.
Since Sandy 4.5 set outperforms Sandy set evidently, I will use Sandy 4.5 set as the default training set.

Figure 4.6 lists the results from several additional comparisons. Here RT_network features refer to both static and dynamic network features from retweet network, and COO_network features refer to corresponding ones from the COO network. Unlike the slight improvement by using retweet features in the user case, here the performance of the network features derived from retweets and mentions are almost the same. I think the possible reasons are: the node level network features are used for user prediction while global level features are used for hashtag prediction, and the variability among node level features are much higher than global level ones, thus node level features seem to able to exhibit the distinction between retweets and mentions more obviously. The second plot shows the results using COO network features. Unlike the retweet network features, the COO network features alone cannot provide good results. The retweet network features have prominent prediction power for smaller number of top hashtags, when the improvement from the baseline ones is also greater. When the number of hashtags involved is large, where many of them are almost inactive (Figure 4.3), the performance of network features degrades and less improvement is observed. Though adding COO features to baseline ones can obtain intermediate improvement, using them alone has the worst performance as always. Moreover, the feature combination with all features only barely beats the combination of basline+retweet features when the number of hashtags is large, but has much worse performance when the number is small. Therefore the COO features are much less effective than retweet features, thus I do not recommend extracting them for the prediction since extra amount of network computing is involved and it leads to low benefit cost ratio. However, when computing resource is not a big concern, they can still be added to provide an alternative solution. As to the text content features, to my surprise, themselves alone are even giving much worse results than the COO features (about 0.1-0.2 on the rank correlation score, not shown in the plot to provide better resolution). What's more, the baseline+content feature combination has no improvement over baseline, and so does the combination of all features over baseline+retweet combination. This simply implies the content features have nearly no predicting power at all, at least to this rank prediction task.

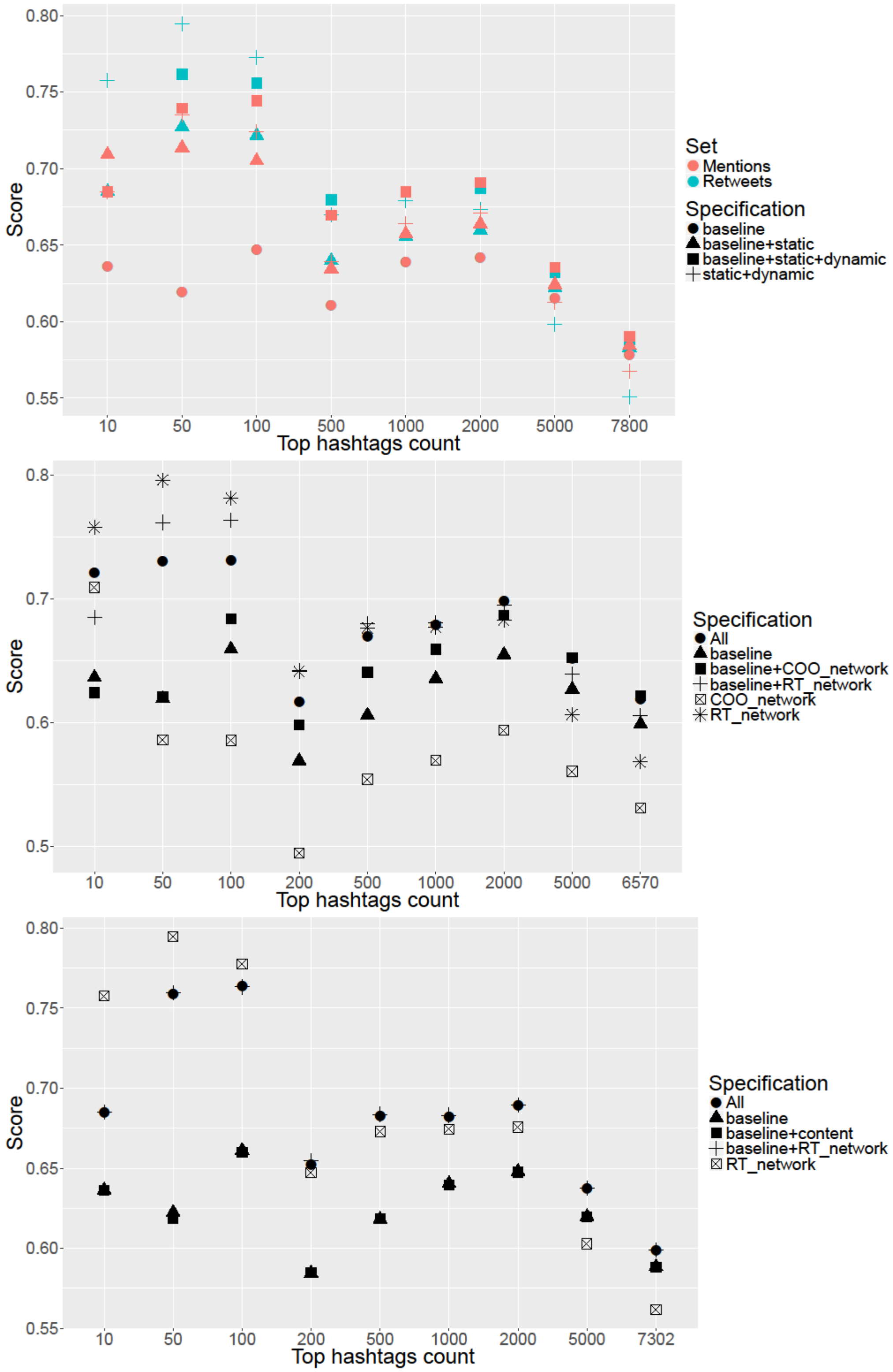


Figure 4.6 Nepal hashtag rank prediction results by varying additional factors

This observation is in contrast to (Ma, 2013), where some of the content features are considered to be effective. I speculate the main reason lies in the context of the prediction task: some derived content features use information from a long term collection of data set with a mixture of all types of hashtags, while my task is focusing on prediction of hashtags restricted to limited time period and a specific event. Regardless of the true underlying reason, I have shown that the retweet network features turn out be much more effective than content features in this special urgent diffusion prediction task. This may have profound implications: content-based features, while could contain rich information and be highly effective potential predictors, cannot fully utilize their potentials and perform well on a prediction task restricted with limited information; while diffusion based network features, on the other hand, can still stably provide decent performance under this difficult condition.

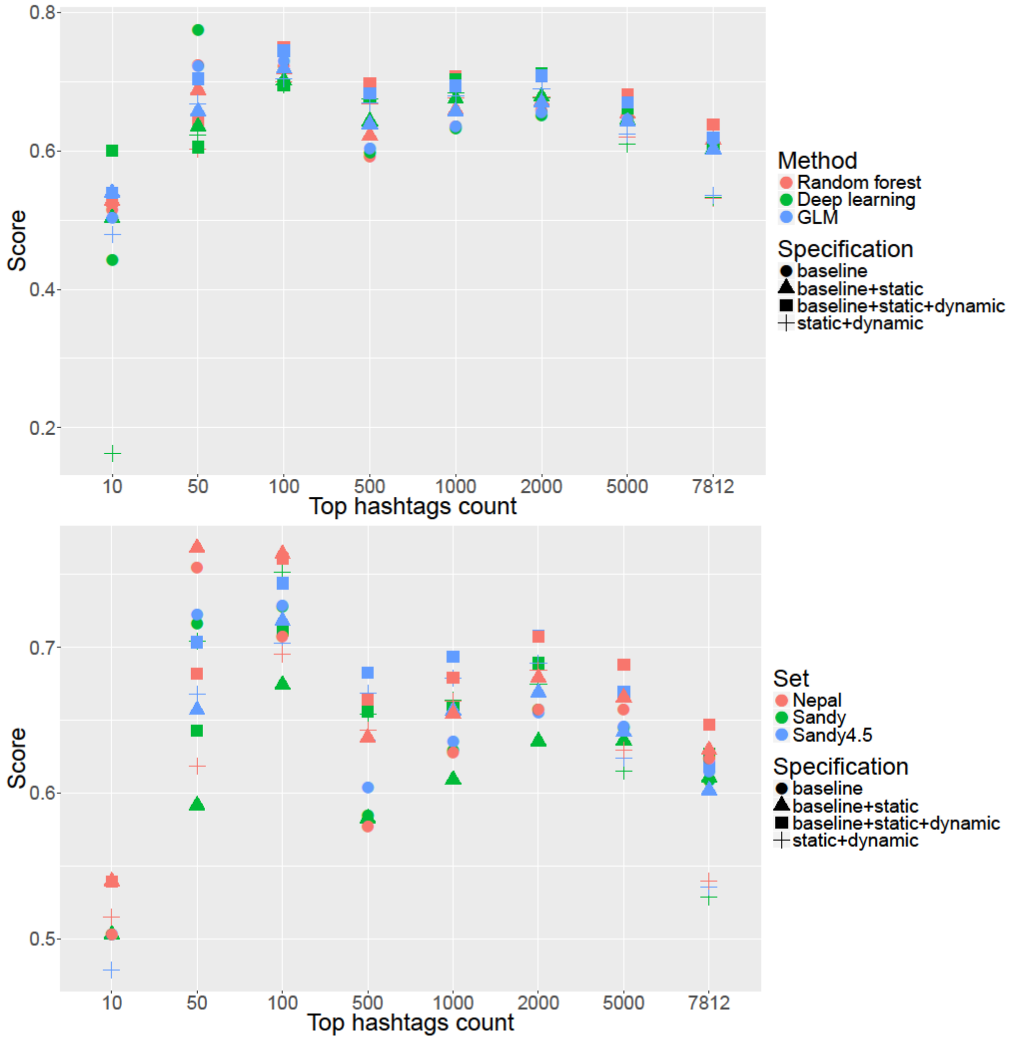


Figure 4.7 Blizzard hashtag rank prediction results by varying methods and training sets

Figure 4.7 shows the results for Blizzard set hashtag rank prediction for extension study and the default Sandy 4.5 training set is used in the first plot. Like in the user case, the results turn to be a little noisier, but these should be reasonable considering the high variability nature of the prediction task. Compared to the other two methods, GLM is still the best choice if I take its various advantages into consideration. The network features still improve the performance overall, just not to the extent of the Nepal case. As to the training sets, Nepal and Sandy 4.5 seem to perform slightly better than Sandy set, but they are close on average. This is in consistent with what I have observed for the user rank prediction—the differences among various training sets are not significant.

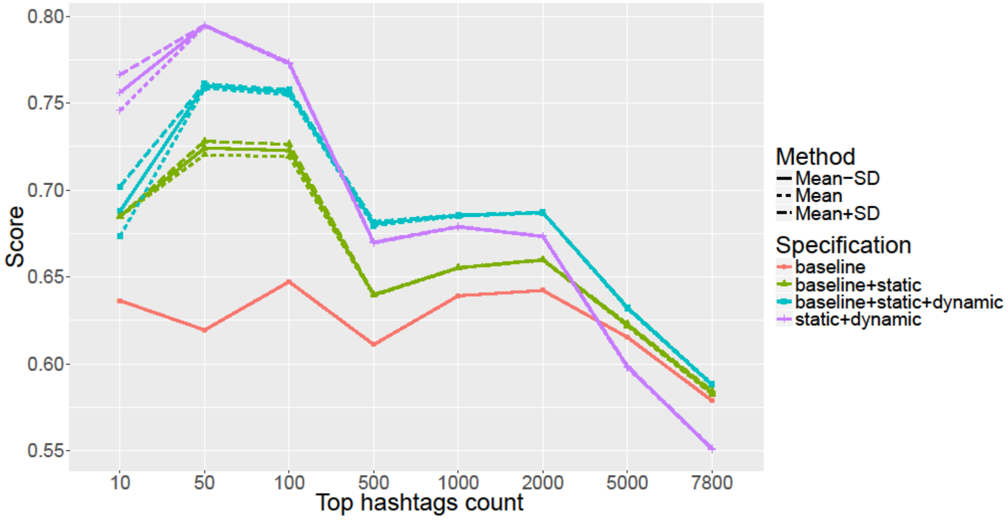


Figure 4.8 Nepal hashtag rank prediction GLM method stability

Same as Figure 3.11, Figure 4.8 is the stability plot for GLM method for the hashtag rank prediction task (n=100). The result is even more stable than the user case, which could be simply because the hashtags have more stable rank. The other two methods still have much worse stability like in the user case and thus are not shown here to save space. Therefore I can say GLM is the best method for both user and hashtag prediction tasks in terms both performance and stability, which could possibly suggest a strong yet stable log-linear dependency between the selected features and the target response.

The results for the classification tasks are in Figure 4.9. For the top quantile classification task the rank prediction has comparable performance as formal classification (same setting as in the user prediction). This is basically the same conclusion I have drawn for the user prediction. However, for the rapidly increasing classification task, rank prediction significantly outperforms formal classification, which is different from the user case. Though not being satisfactory based on the PRAUC score, the performance of the rank prediction is nearly 8 times against the random guess across different increasing percentages, compared to only about 2 times in the user case. This improvement over user prediction may also result from the more concentrated hashtag response distribution. Recalling the previous quantile analysis, the same response threshold bar—10 for the rapidly increasing classification corresponds to about 80% quantile in the hashtag case and about 60% quantile in the user case respectively. Therefore the positive data points in the hashtag prediction case contain higher fraction of top-tier points by response than that of the user prediction case, which could lead to better alignment of the classification with rank prediction in the hashtag case over the user case.

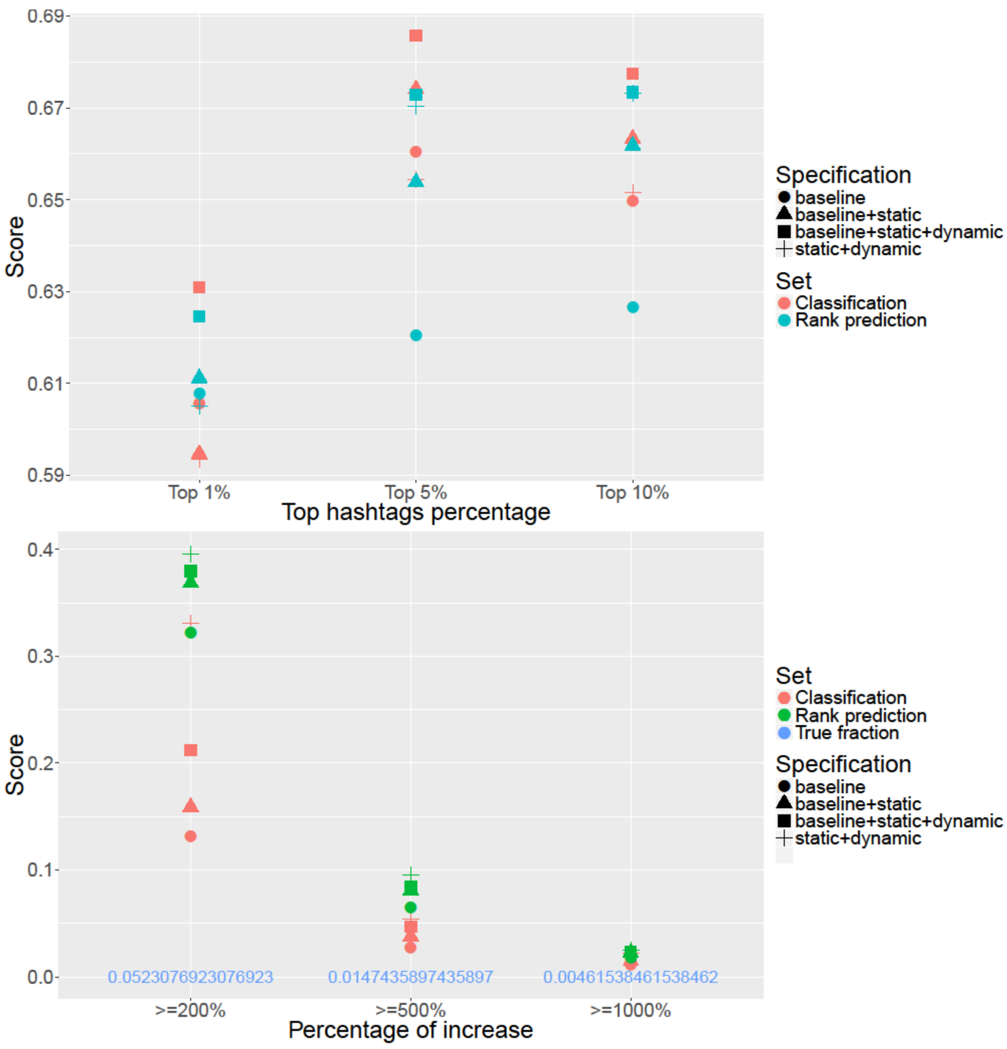


Figure 4.9 Nepal hashtag classification results

Figure 4.10 shows the variable importance ranking for the GLM method. Same to the user case, I am using the baseline+static+dynamic feature combination. But this time I reverse the order in the frequency plot since most of the features involved have been selected every time (n=100), more than the user case, which also explains the better method stability for the hashtag prediction. As to the top 10 variable magnitude rank, similar patterns as the user case are observed: there are constantly important baseline features like number of tweets and URLs, and also some varying dynamic network features such as the WCC values within different time windows, where the fractions of the selected baseline features and network features are nearly 1:1. Interestingly, for dynamic network features derived from the same network statistics—WCC, they can even have different signs in the weights; moreover they are also significant regardless of the signs. This again illustrates the necessity of adopting the dynamic network features: they can show varying effects under varying time intervals and each of them may turn out be significant as well; while simply using static features will not only hide these dynamic patterns but degrade the performance as well.

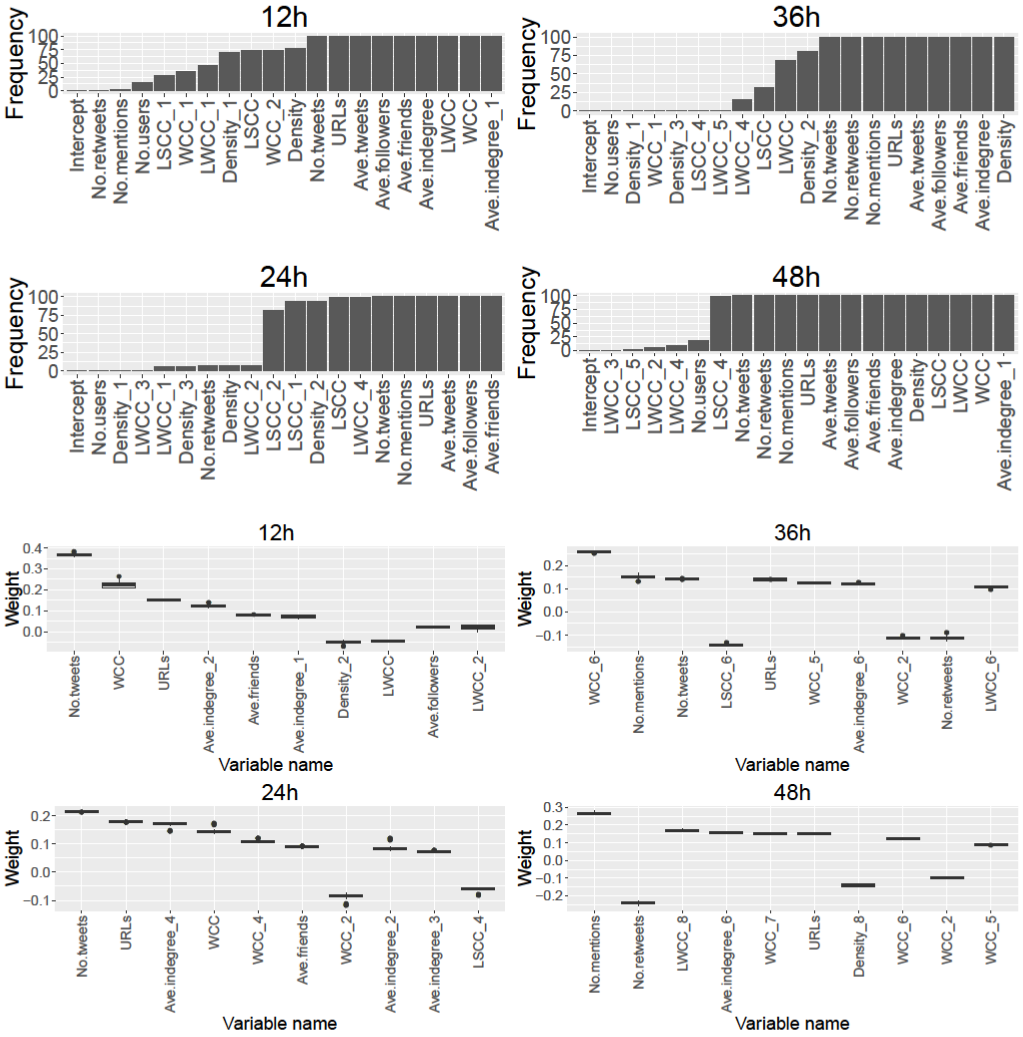


Figure 4.10 Nepal hashtag rank prediction GLM method variable importance

4.2.4 Direct prediction results

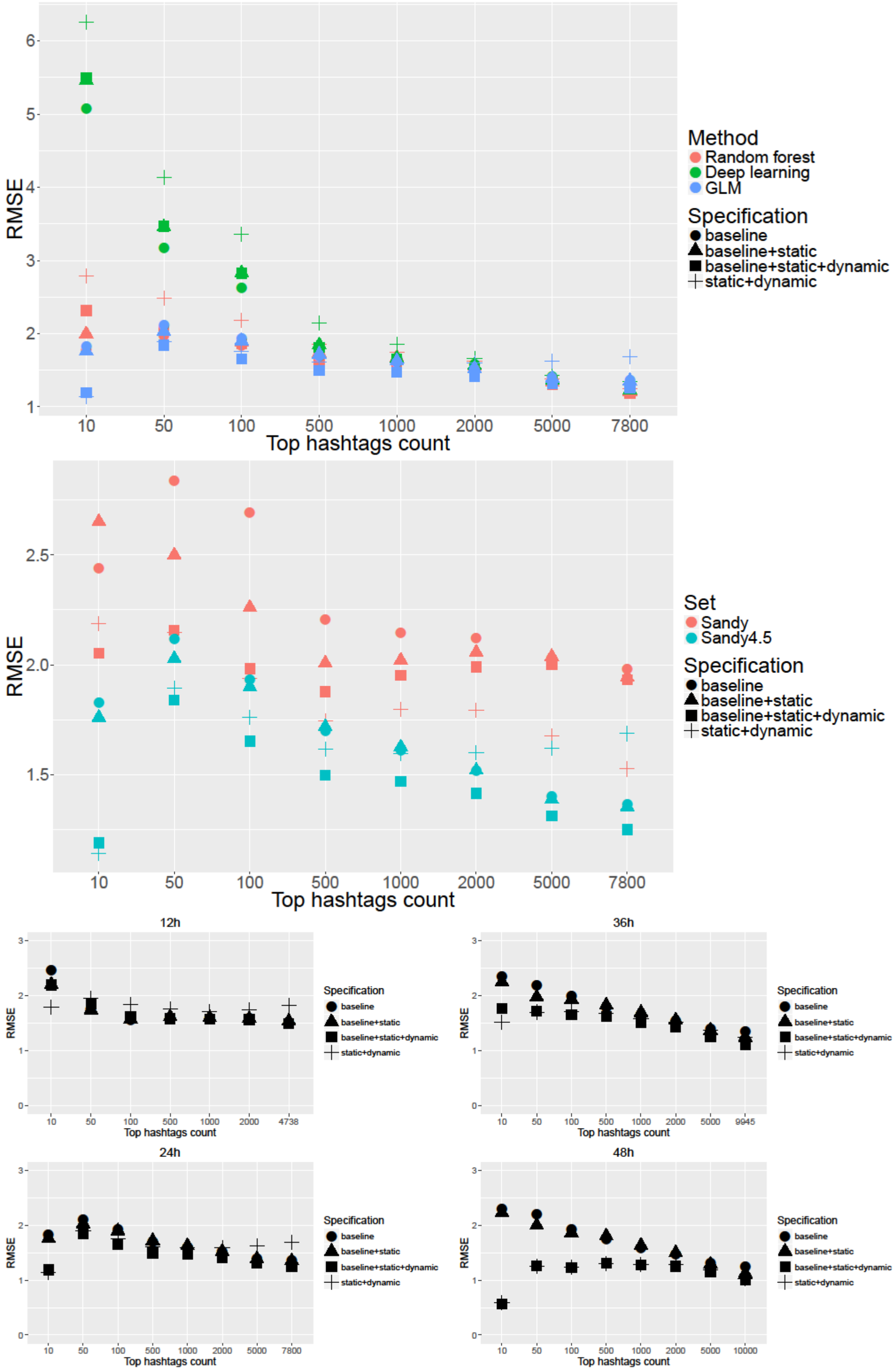


Figure 4.11 Nepal hashtag direct prediction results varying by methods, training sets and T

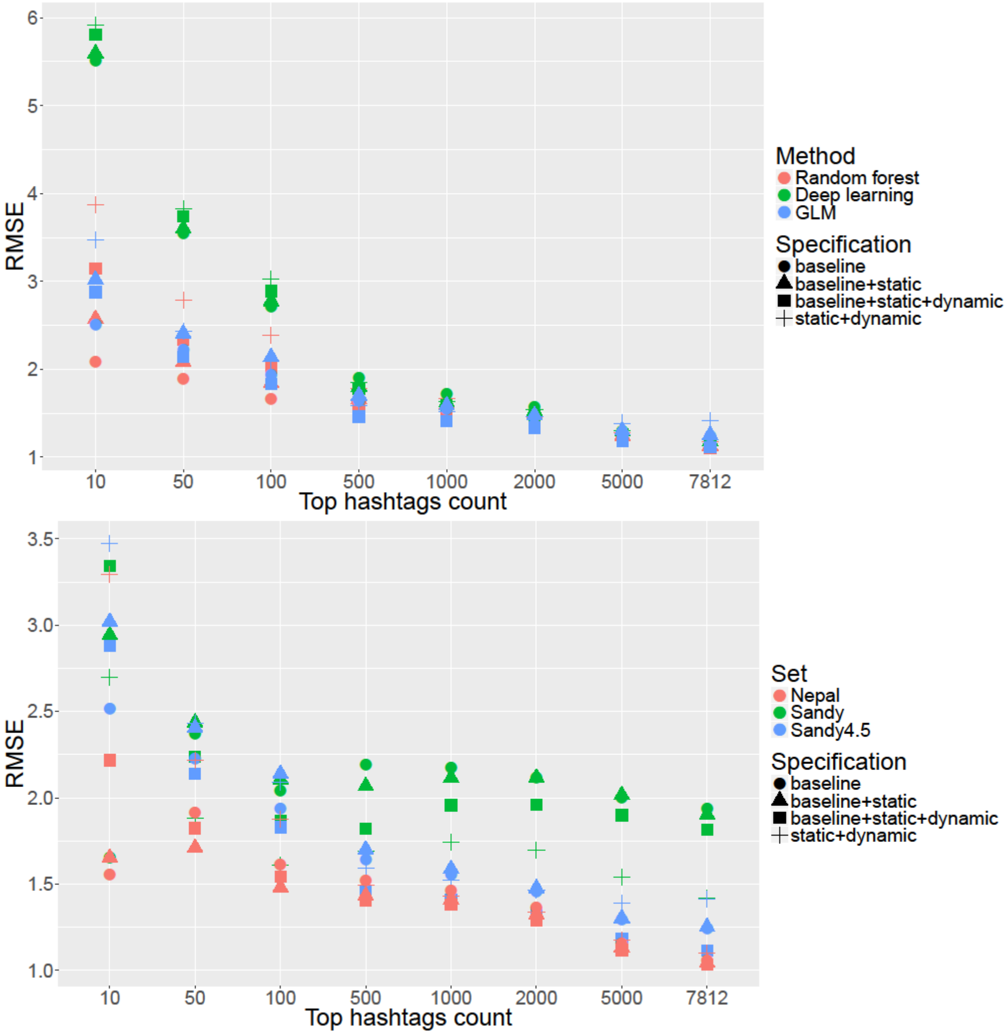


Figure 4.12 Blizzard hashtag direct prediction results by varying methods and training sets

Following similar settings as the user case, Figure 4.11 and Figure 4.12 show the direct prediction results for Nepal set and Blizzard set respectively. Compared to the user case, hashtag direct prediction exhibits both similarities and distinctions. In both cases, on average across all settings GLM has the best performance and addition of network features make improvement on prediction over baseline features. Also the variations among methods and training sets are larger than the rank prediction. But unlike to the user prediction case, the variations among training sets are much smaller for hashtag prediction, which demonstrates more stable prediction. A possible explanation is that the heterogeneity among hashtag features and response across various data sets is much smaller compared to that in the user case. Although the selection of training sets is of less concern than the user case, prominent improvement, which is even larger than improvement from good features selection, can still be obtained if a proper choice is made.

Set	Sandy	Sandy 4.5
Log scores (top 3000)	0.722	0.746

Set	Sandy	Sandy 4.5	Nepal
Log scores (top 3000)	0.643	0.810	0.140

Table 4.3 KS statistics of past scores for various training sets with Nepal set (top) and Blizzard set (bottom)

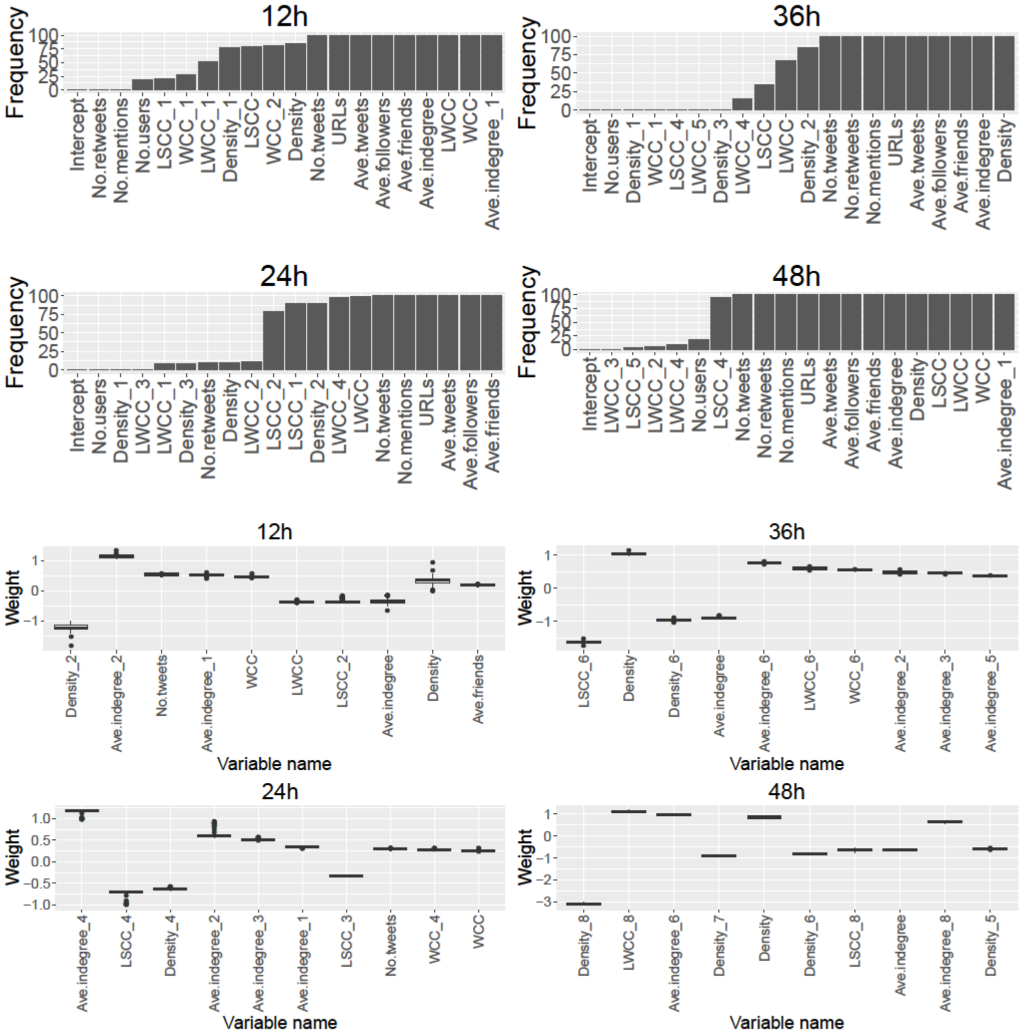


Figure 4.13 Nepal hashtag direct predictions GLM method variable importance

In Table 4.3 the KS statistics from past scores for training sets selection are presented. Since now different sets have different number of data points (see Table 4.2), the comparison cannot be made directly with different sample sizes. Here I only use the top 3000 data points for the KS statistics calculation. For the Nepal set prediction, the two values obtained are close, thus I cannot make a good decision on which training set is better. For the Blizzard set prediction, though I still cannot distinguish whether

Sandy or Sandy 4.5 is better, I can confirm with high confidence Nepal set should be the best choice. And the result indeed fits this well. Thus I believe the KS statistics with past scores can still be a good empirical criterion for training sets selection, given the distinction among the statistical values is large.

Figure 4.13 displays the variable importance for direct prediction task for Nepal set using GLM method. Generally the plots look similar to Figure 4.10: most features are selected by the model all the time and top ranked features consist of both baseline and network features. One thing slightly different is that I observe higher fraction of dynamic network features in the top 10 ranked features by weights, as far as to 8-9 out of 10. This demonstrates the dynamic network features are playing more important roles for the direct prediction task. And greater improvement effect can be observed for hashtags staying in the top rank, when these features are really taking meaningful values instead of many nulls.

Hashtag (rank/log(score+1))	Past score	Predicted by SD	Predicted by B	Predicted by BSD	True
rebuild	**13/7.80**	**458/3.14**	**15/6.68**	**95/5.17**	**1017/3.30**
supportnepal	138/5.29	56/4.75	77/5.06	43/5.65	13/8.72
earthquakeagain	50/6.31	37/5.53	66/5.12	52/5.43	15/8.49
helpnepalchildren	**5970/0.69**	**7322/-1.29**	**4637/0.17**	**5887/0.15**	**9/9.31**
3news	**4926/0.69**	**7032/-1.29**	**6421/-0.22**	**6961/0.006**	**18/8.34**

Table 4.4 Prediction further investigation on selected hashtags using GLM

Missing counts by rank	Missed by SD	Missed by B	Missed by BSD	Missed by all	Count of the union
Top 10	3	3	3	3	12
Top 20	7	8	7	7	24
Top 50	21	23	22	17	66
Top 100	39	44	40	31	139

Table 4.5 Missing hashtags count under various feature sets using GLM

Following the same ideas as Table 3.5 and Table 3.6, I further investigate the effect of features on prediction based on several individual hashtags. Tracing back SD represents feature combination static+dynamic, B represents baseline and BSD represents baseline+static+dynamic; the values in the table are rank and log(score+1) respectively. In Table 4.4, the hashtag 'rebuild' is over ranked by B but pretty well predicted by SD if I look at the direct prediction score vs. the true score. It is interesting to see here for this hashtag with log past score 7.8, the feature combination SD only predicts 3.14 compared to 6.68 provided by B, which clearly indicates there is unique information included in those network features so that they are suggesting a different result. For hashtags 'supportnepal' and 'earthquakeagain' the results are very similar for all features combinations and past scores, but they are different from the true values. This illustrates that for hashtags with intermediate past activity but with high future activity, none of the features may be able to make accurate prediction. The situation is much worse for hashtags 'helpnepalchildren' and '3news', where they showed barely activity in the past but high activity in the future, and none of the features can make even reasonable prediction. This is exactly the same as I have noticed in the user case: I cannot make good prediction out of nothing. But generally speaking, similar to the user case, the BSD combination will provide the most reliable and stable results since it can make use of the advantages of both network and baseline features. Table 4.5 serves the same purpose as Table 3.6 to provide a suggestion for maximum top hashtags inclusion. The improvement by union all results seems to be less, but it still depends on the actual cost function to determine whether it is worth to do or not.

4.3 Prediction on latent topic popularity

4.3.1 Features and response

Following the same settings as mentioned earlier, I consider hashtags as documents to build the LDA topic model with number of topics to be 10. More specifically, I am using all hashtags (after content filtering) within period [0,T] to construct the LDA model, and then use the model to infer the topic probabilities for each hashtag involved. The weighted aggregation (the weights are the corresponding probabilities) of features and response of hashtags will be used as features and response for each topic. And since the content of documents (hashtags) will drift and change over time (will be shown later), the topics formed from them will also prosper and decay relatively fast. Thus I will set the default prediction period h to be 2d instead of 5d. The weights of features are given by the LDA model trained within period [0,T], while the weights for the response are inferred using the trained model on hashtags text within period [T,T+h]. For consistency consideration, I need to maintain the same set of hashtags for derivation of features and response. There may be newly invented hashtags in period [T,T+h] that have good alignment with existing topics, but I have to exclude them from the prediction task since they do not preserve features in the training period. The potential bias introduced by this exclusion is inevitable in this prediction task setting. Moreover there are other factors to be considered like documents drifting and noisy hashtags filtering, I will only select the top 10k hashtags by retweets during [0,T] for features and response extraction, which is the same thing as I have done for the user and hashtag prediction tasks. Thus the response—the topic popularity will be the weighted aggregation of top 10k hashtags response with weights inferred by the existing LDA model upon the text of the same 10k hashtags during period [T,T+h].

As to topic features, they also include the baseline features and network features, with the corresponding weights are from the LDA model during period [0,T]. The baseline features will simply come from the hashtag baseline features plus one additional 'Average polarity', which I believe should be a useful feature for topics. For baseline features that are defined as summations: 'Number of tweets', 'Number of retweets', 'Number of mentions', 'Number of tweets with URLs', 'Number of users', I use the weighted aggregation of hashtags ones to form the topic ones. For others that are defined as means: 'Average user tweets', 'Average user followers', 'Average user friends' and 'Average polarity', I use the weighted average of hashtag values as topic ones. But for network features I cannot do the same since network features are defined on one whole network and weighed aggregation of network statistics from various networks have no practical meanings. In this case I can construct a weighted network for each topic, with the edges coming from all edges of top 10k hashtags and the weights are corresponding probabilities from each hashtag for the same topic. Since there are not many well defined network statistics for weighted directed networks, I introduce two network statistics for this weighted network: 'Average indegree' and 'Median outdegree'. Other parameters not mentioned will simply adopt the default settings as previous sections.

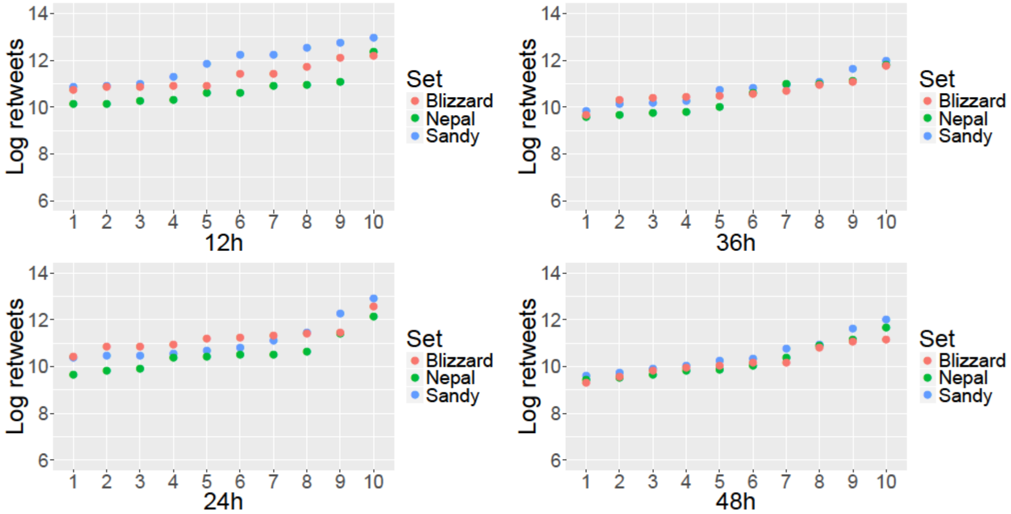


Figure 4.14 LDA topic response comparisons across various data sets

Figure 4.14 lists the topics popularity (response with h=2d) over various T values across three data sets. Here the topic indices are simply ranked by increasing order of the topic response for better alignment comparison. The default topic indices are ranked by the topic coherent scores, which are the ones used in Figure 4.15. From the plots both within the sample range and cross samples (data sets) range of the topic response are about 2 on the natural logarithm scale.

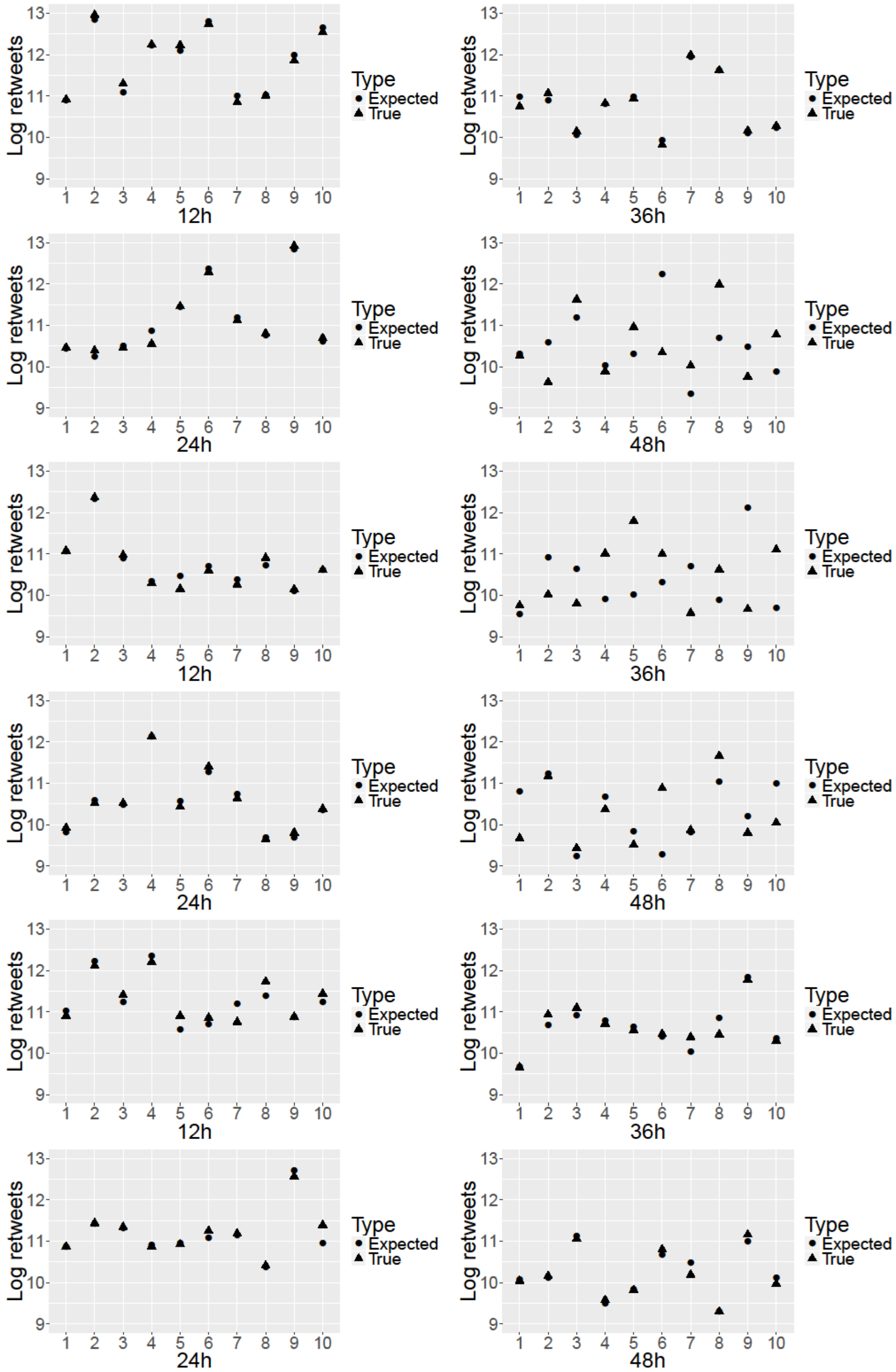


Figure 4.15 LDA expected versus true topic response comparison across various data sets (the data sets are Sandy, Nepal and Blizzard from top to bottom respectively)

Since both features and response are weighted aggregation of hashtags values, they tend to have more homogenous within sample and cross samples distribution (only the distribution of the response is shown here). This is different from both user prediction and hashtag prediction, where the distribution is heterogeneous in either aspect. Thus for the topic prediction task I can simply adopt the direct prediction approach, where the unfavorable effect of heterogeneity is reduced significantly.

Topic_5			Topic_9			Top 10 hashtags	
tokens	probs_expected	probs_true	tokens	probs_expected	probs_true	name	volume
'earthquak', 'help', 'quak', 'aid', 'toll', 'death', 'relief', 'effort', 'peopl', 'victim'	0.0537	0.279	'member', 'quakehit', 'com', 'click', 'hat', 'church', 'coordin', 'morto', 'preacher', 'return'	0.341	0.0299	nepal	166411
	0.0282	0.367		0.544	0.0220	nepalearthquake	59348
	0.0485	0.373		0.541	0.0216	earthquake	23644
	0.0362	0.406		0.549	0.0202	nepalquake	20105
	0.0460	0.287		0.540	0.0322	nepalquakerelief	11610
	0.0411	0.145		0.361	0.103	helpnepal	11450
	0.0671	0.579		0.633	0.0686	helpnepalchildren	10438
	0.0405	0.237		0.245	0.0240	prayfornepal	6709
	0.0322	0.0315		0.719	0.0279	supportnepal	5569
	0.0449	0.389		0.545	0.0466	msghelpearthquakevictims	5215

Table 4.6 Selected topic probabilities comparison for Nepal set with T=36h

Figure 4.15 shows the response comparison using two types of weights (probabilities): the expected weights—ones indicated by the training model during period [0,T] and the true weights—ones derived from the text of same top 10k hashtags during [T,T+h]. The motivation behind is to investigate the document content changing over time. From top to bottom the sets are Sandy, Nepal and Blizzard respectively, and the title of each subplot corresponds to the T value. For majority of the topics across various T values and data sets, the expected values are close to the true values, which indicates the content of the corresponding documents is relatively stable at least to a short term future. But there are a few cases when the discrepancy is big—seen in the 48h plot for Sandy set, the 36h and 48h plot for Nepal set. These cases imply the underlying hashtags (at least some major ones) are experiencing drastic changes from present to the near future since all topics are from existing models and they will remain relatively static. Two topics that have exhibited the largest discrepancy in the Nepal 36h plot are further studied as examples to investigate the document drifting effect, and the results are listed in Table 4.6.

Index	Top 10 tokens
1	'que', 'por', 'del', 'con', 'sandi', 'los', 'york', 'las', 'para', 'huracan'
2	'sandi', 'school', 'hurrican', 'tomorrow', 'power', 'day', 'get', 'class', 'cancel', 'like'
3	'birthday', 'chill', 'candl', 'tormenta', 'cooki', 'social', 'cheek', 'cuba', 'cours', 'tras'
4	'blew', 'reason', 'dog', 'sandi', 'gym', 'roof', 'wors', 'dear', 'hate', 'calm'
5	'sandi', 'hurrican', 'bitch', 'safe', 'east', 'coast', 'everyon', 'hope', 'fuck', 'stay'
6	'help', 'hurrican', 'sandi', 'donat', 'obama', 'victim', 'relief', 'romney', 'amp', 'via'
7	'hurrican', 'sandi', 'close', 'due', 'amp', 'power', 'the', 'new', 'storm', 'wind'
8	'sandi', 'walk', 'someon', 'run', 'bore', 'alright', 'gas', 'drink', 'hurrican', 'water'
9	'anyth', 'spongebob', 'sandi', 'badai', 'hair', 'swim', 'twerk', 'whore', 'dan', 'hoy'
10	'sandi', 'hurrican', 'like', 'get', 'the', 'power', 'eat', 'came', 'room', 'are'

Index	Top 10 tokens
1	'yoga', 'des', 'earthquak', 'dog', 'south', 'auction', 'rescu', 'wow', 'everest', 'surviv'
2	'voor', 'per', 'van', 'een', 'met', 'uit', 'aan', 'het', 'aardbev', 'che'
3	'todo', 'mit', 'ein', 'den', 'bruce', 'les', 'til', 'anim', 'pour', 'ist'
4	'der', 'die', 'und', 'erdbeben', 'nach', 'das', 'aus', 'von', 'club', 'auf'
5	'rais', 'fund', 'earthquak', 'fundrais', 'relief', 'help', 'donat', 'money', 'today', 'support'
6	'terremoto', 'que', 'los', 'por', 'del', 'para', 'con', 'las', 'muerto', 'ayuda'
7	'earthquak', 'quak', 'toll', 'death', 'hit', 'miss', 'helicopt', 'anoth', 'rescu', 'dead'
8	'gempa', 'korban', 'indonesia', 'untuk', 'wni', 'ronaldo', 'bantuan', 'tim', 'cristiano', 'dari'
9	'help', 'donat', 'earthquak', 'peopl', 'pray', 'support', 'pleas', 'need', 'prayer', 'victim'
10	'earthquak', 'relief', 'help', 'effort', 'aid', 'via', 'disast', 'donat', 'quak', 'proud'

Table 4.7 Top 10 tokens for each topic from Sandy set (top) and Nepal set (bottom)

In Table 4.6 the probabilities for top 10 hashtags by retweet volume are shown to reflect the content variations of the topics. Clearly topic 5 is underestimated while topic 9 is over estimated. This suggests the content of underlying hashtags is shifting from topic 9 to topic 5, where I can distinguish their differences by the top 10 tokens. From this situation I can perceive the differences between the topics defined simply by hashtags and by latent topic models. While the former are annotated by a variety of people who believe the content should be relevant to a concentric but relatively vague idea, the later are solely content based and judged by computers using the distribution of tokens. The ideal topic model should be able to reach a consensus with human belief to some extent, with the extent can also be finely tuned.

To further explore the topics formed by the models, I list the top ranked tokens for each topic formed by the full data sets in Table 4.7. There many of the topics indeed delivering concordant information, such as topic 2,5,6,7 for Sandy and topic 5,7,9,10 for Nepal. Though there are also topics formed by languages other than English and thus cannot be understood well, this can barely be avoided since I do not have a well labeled training set to perform the language classification task. Generally speaking, the LDA model is really generating meaningful topics, although the best number of topics is still a concern here—larger number of topics can either split more general topics into smaller but also more concentric ones or capture some hidden interesting small topics, while smaller number of topics can aggregate fragmented yet less meaningful topics into consistent and more meaningful ones. But as mentioned earlier, the main goal of my study is not trying to build good topic models, but to investigate and compare how the prediction task will vary from the two topic definitions—hashtags and latent LDA topics. Thus the quality of the topics is of less concern and should not affect my final conclusion.

With the number of topics fixed, there will be drastic topic drifting over time due to content variation of the underlying hashtags. Some formed topics may only exist within a short period of time but some others may persist in long term. In Table 4.8, I select some persistent yet meaningful topics to see what kind of topics survives longer periods. The first two topics are from the Sandy set and the last one is from the Nepal set, the boldfaced tokens are ones that are shared through different periods. Here I choose the most persistent yet meaningful topics by measuring the fractions of overlapping tokens within the top 10 tokens within different periods since topic indices tell nothing about their relationship, this criterion can be modified subject to personal preference and I am simply using it for illustrative purposes. Again the indices of the topics correspond to the coherent ranking, thus they can serve as a rank of topic quality to some extent. The first Sandy topic expresses a clear theme: 'Hurricane Sandy is approaching east coast and hope everyone stay safe'. This one should be the most long-lasting yet consistent topic observed for Sandy set, which should be reasonable considering the particular feature of events related to natural disasters. The second Sandy topic appears less straightforward but I can still infer the central idea is about school class cancellation due to the hurricane, where the discussion is likely to be initiated and prevailed among students.

The Nepal topics seem to less persistent and the best one I have found only has 4 tokens in common through all periods. But the idea conveyed is clear: all about relief, donation and aid due to deaths during the earthquake. There are other less persistent but maybe more centric topics which are also worth to study—about more specific issues such as certain rescue actions or church prayer activities. Therefore the topic model is achieving meaningful performance, and my main goal is to predict the future popularity of these topics.

Collection period	Index	Top 10 tokens
12h	9	**'safe'**, **'sandi'**, **'hurrican'**, **'everyon'**, **'stay'**, **'coast'**, **'east'**, **'hope'**, 'power', 'pray'
24h	5	**'safe'**, **'sandi'**, **'hurrican'**, **'everyon'**, **'coast'**, **'east'**, **'hope'**, **'stay'**, 'peopl', 'pray'
36h	7	**'safe'**, **'hurrican'**, **'sandi'**, **'east'**, **'coast'**, **'everyon'**, **'stay'**, **'hope'**, 'peopl', 'pray'
48h	8	**'hurrican'**, **'sandi'**, **'east'**, **'safe'**, **'coast'**, **'everyon'**, **'stay'**, **'hope'**, 'new', 'affect'
All	5	**'sandi'**, **'hurrican'**, 'bitch', **'safe'**, **'east'**, **'coast'**, **'everyon'**, **'hope'**, 'fuck', **'stay'**

Collection period	Index	Top 10 tokens
12h	10	**'school'**, **'sandi'**, **'tomorrow'**, 'cancel', 'class', **'hurrican'**, 'day', **'get'**, 'power', 'work'
24h	8	**'sandi'**, **'hurrican'**, **'school'**, **'get'**, **'tomorrow'**, 'sleep', 'like', 'fuck', 'bitch', 'cancel'
36h	10	**'sandi'**, **'hurrican'**, **'school'**, **'tomorrow'**, **'get'**, 'fuck', 'bitch', 'day', 'got', 'class'
48h	10	**'school'**, **'tomorrow'**, 'class', **'sandi'**, 'cancel', **'hurrican'**, 'day', **'get'**, 'thank', 'power'
All	2	**'sandi'**, **'school'**, **'hurrican'**, **'tomorrow'**, 'power', 'day', **'get'**, 'class', 'cancel', 'like'

Collection period	Index	Top 10 tokens
12h	1	**'earthquak'**, **'effort'**, 'toll', 'call', **'quak'**, 'peopl', 'death', 'suppli', **'help'**, 'aid'
24h	4	**'earthquak'**, **'help'**, 'relief', 'toll', 'donat', **'quak'**, 'peopl', 'death', **'effort'**, 'rise'
36h	5	**'earthquak'**, **'help'**, **'quak'**, 'aid', 'toll', 'death', 'relief', **'effort'**, 'peopl', 'victim'
48h	8	**'earthquak'**, **'help'**, 'toll', **'quak'**, 'death', 'relief', 'aid', 'donat', 'peopl', **'effort'**
All	10	**'earthquak'**, 'relief', **'help'**, **'effort'**, 'aid', 'via', 'disast', 'donat', **'quak'**, 'proud'

Table 4.8 Selected persistent topics over various periods for Sandy and Nepal sets

While the true future popularity of previous topics is hard to find, I use the weighted aggregation of hashtag popularity as the approximation, and it is the best available approach given the data and information in hand.

4.3.2 Prediction results

The direct prediction is expected to lead to relatively good results and it indeed does. Figure 4.16 shows the direct prediction results for Nepal set varying T and methods. The RMSE is below 1 on the log scale for the best setting, considering the corresponding magnitude for the response is about 10~12, I believe the results are satisfactory. For the prediction methods, the GLM no longer performs the best since the prediction results are highly unstable. The main reason should be the singular fitting problems that occur occasionally. Moreover, warning messages are received to remind there are too few samples for the CV—the minimal requirement is 3 for each fold. But even the number of folds changes from the default value 10 into 3, the singular fitting issue is still there. There are two driving factors for this: the sample sparsity and the homogenous features distribution, where the latter one could be more fatal to the linear model. Given more samples, this issue may be remedied to some extent by CV and feature selection, but it is not feasible in this study due to the setting of my problem. Thus GLM may not be a good choice for this prediction task.

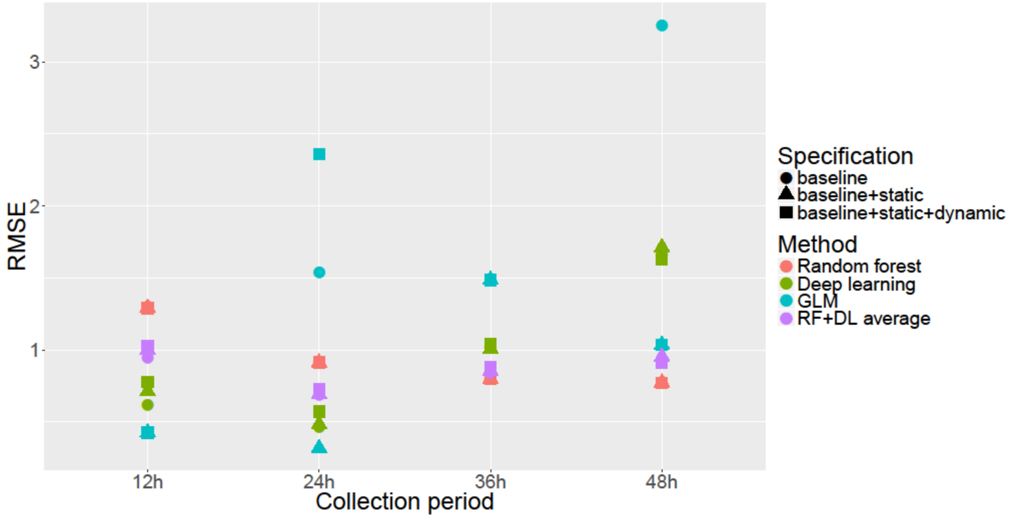


Figure 4.16 LDA topic popularity predictions for Nepal set varying T and methods

In contrast, the other two nonlinear methods perform relatively stable: with DL performs better during the first two periods and RF performs better during the last two. To achieve the best stability, I suggest the approach using the average scores of RF and DL methods, which gives more stable results staying below a RMSE of 1 consistently. As to the feature sets, generally I do not see any significant improvement over baseline features. Given the great computational cost (mainly for the weighted graph construction), network features are not recommended for this task since the baseline ones already provide satisfactory results.

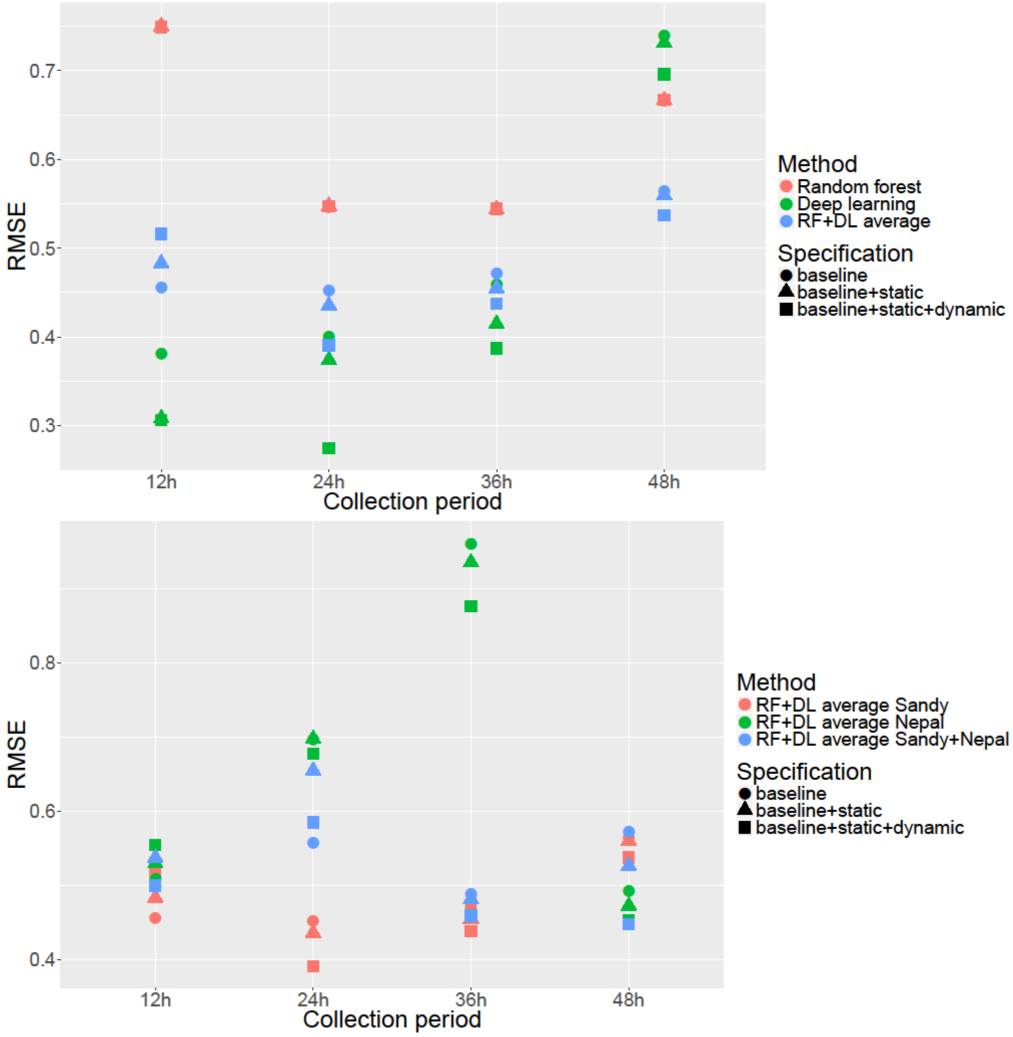


Figure 4.17 LDA topic popularity predictions for Blizzard set varying T, methods and training sets

Figure 4.17 shows the topic prediction results for Blizzard set with GLM method removed. The results are even better than the Nepal case and RF+DL average also gives the most stable performance with RMSE around 0.5. For this case, the network features demonstrate some improvement over baseline ones, especially for the DL approach. However, as I mentioned earlier, it is still not necessary since the overall performance of all feature sets is already good. The second plot shows the results for the RF+DL average approach with varying training sets. The Sandy set performs best on average while Nepal set does the worst on average. Although the general performance is still good, the variability among methods and training sets are much more than the feature sets, which seems to be natural when it comes to direct prediction instead of rank prediction. Thus, choice of training sets proves to be much more important than choice of features in this case. The empirical measure for training sets selection I have used before—the KS statistics may not be a suitable one here since the sample size is too small to make the comparison results powerful enough. Then a proper alternative would be to simply use the combination of the two sets: as shown in Figure 4.17, the combination can give relatively stable results even though no further information about which one to choose is provided.

4.4 Discussion

4.4.1 Summary

For the hashtag prediction part, I intentionally follow nearly the same settings as the user prediction to identify their similarities and differences. It is easy to observe that they should share much in common from each other since both the user activity and hashtag activity will largely depend on single tweet popularity (number of retweets it obtains); especially for tweets with extremely high popularity, the corresponding users or hashtags (if there is any) will also be attached with large scores as well. More specifically, I have found the following common aspects:
1. The response distribution for both the users and hashtags followed well with the power law distribution, demonstrating that it is a general phenomenon for information diffusion on Twitter regardless which level to look at: tweets (looking at the retweets distribution of each tweet), users or topics.

2. The network features, especially the dynamic ones, significantly improve over baseline ones across different prediction tasks for both cases. To view this more generally, the additional dynamic information included in those features is indeed helpful in predicting the near future behavior, for both users and hashtags.

3. For both cases the GLM method not only gives the best on average performance, but also possesses other advantages over the other two non-linear methods: such as stability, run time efficiency and well defined variable importance etc. Similar to the user case, there is also a strong log-linear dependency between the selected features and the response. The fact that no better performance is observed from the two nonlinear methods towards to the linear method again shows there is no significant nonlinear relationship between the selected features and the response. More generally, this can imply there is strong log-linear dependency (while no significant nonlinear dependency on the log scale) between past activity and near term future activity, where the activity could be quantified by various metrics (number of retweets, number of mentions, URLs etc.) based on different levels (tweets, users, topics etc.).

4. Given the heterogeneity among the data sets, it is not surprising to see the rank prediction turns out to be more stable than direct prediction in both situations. And the rank prediction results can be directly used for top-tier users or hashtags classification as well. As to the direct prediction, the choice of training sets is a crucial issue to be considered in both cases, and the KS statistics proves to serve as a good empirical measure for both tasks.

5. Since both the baseline and network features used in the prediction tasks rely heavily on past behavior, they will be ineffective for samples with low past activity, regardless the samples are users or hashtags.

Provided with the above common points, there are also distinctions between the two scenarios:
1. The "Matthew effect" for hashtags is much stronger than for that of users. This can be seen directly from the quantile distribution plots, where the distribution is more extreme for hashtags than users, and their maximum values are differed by about 2 orders of magnitude.

2. Perhaps due to the more extreme distribution, the prediction results exhibit better stability than the user case. For the rank prediction, on one hand the GLM method stability plots illustrate better stability for hashtag prediction directly; on the other hand, a larger fraction of variables are selected by the GLM model in hashtag prediction than user prediction, which can also be evidence of better stability. For the direct prediction, the distinction is more obvious. Wrong choice of training data can lead to totally absurd results in the user case, while still remain relatively reasonable ones in the hashtag case. A possible explanation for this might be: because of the more extreme distribution for all data sets, the distortion due to training and testing sets heterogeneity is somehow reduced—the distributions become more similar from each other in the log scale.

3. Much better results exist in the rapidly increasing classification task for hashtags. The underlying reason may also be the more extreme distribution, where more rapidly increasing samples belong to top-tier hashtags and exhibit higher activity, thus can lead to much improved results with network features.

For the latent topic prediction, I consider it as a weighted version of hashtag prediction: the features and response are all weighted aggregation of hashtag ones. As to the topics, the LDA model is doing a decent job to pick up some concentric and persistent topics, although the quality of topics still have much room for improvement. The way to quantify the topic popularity may be biased and inappropriate, but it is the currently best approach available to me given the information I have. Due to the weighted aggregation, the distribution of features and response become much more homogenous. This makes standardization unnecessary and thus I can just go ahead with the direct prediction tasks. The followings are the main discovery for the latent topic prediction:
1. Generally speaking the prediction results are good in terms of RMSE measure, which should not be surprising given the good alignment of topic response in Figure 4.14. The GLM method no longer performs well due to unstable results led by the singular fitting problem. Increasing the sample size could fix the problem but it is not feasible in my setting. The other two nonlinear methods are stable and using the average scores of RF and DL methods is the most stable approach.

2. The prediction improvement of network features over baseline ones is limited. It is likely because these network features do not have good differential power among the topics, which is a result of the homogenous distribution. Consideration the computation cost of weighed graph construction and network features extraction, I do not recommend using these features for this task since the performance of baseline features is already good, unless better accuracy is highly preferred and rich computation resources are granted.

3. The performance variations due to methods and training sets are much larger than choice of features, which is similar to my observation for the user and hashtag direct prediction tasks. But I do not think KS statistics is a good choice for training sets selection for this task due to the small sample size and limited testing power. I suggest using the combination sets for better stability purpose if no other information about the better training set is provided.

4.4.2 Future work

Similar to the user prediction tasks, I would like to extend the analysis to other types of events and platforms. Combined with the diffusion tracking and user prediction analysis, this topic prediction task could also be one function for the more general-purposed event diffusion study user API granted with enough time and engineering effort. Given more time for investigation to the latent topic part, I would like to construct an online framework that can fulfill and concatenate the following tasks well: topic formation or detection, topic quality identification, topic evolution monitor and topic popularity prediction.

Chapter 5: Conclusion

5.1 Discoveries and contributions

For urgent event diffusion through Twitter, my primary assumption in this study is that the diffusion of urgent events mainly relies on retweeting behavior. There are definitely other effective ways for information diffusion to occur: such as posting new relevant tweets, adopting hashtags or URLs, and mentions. But as discussed before, retweets demonstrate several advantages over the other measures: they not only provide solidly tractable path for the diffusion (compared to normal tweets), but also are stronger indictors of information diffusion (compared to mentions) and include more general forms of underlying tweets (compared to tweets with hashtags or URLs). Therefore they are the best candidate for the study of information diffusion. Based on this, the study of urgent event diffusion can be considered as study of the underlying retweet network, and it is the main reason why I emphasize much on network statistics in my study.

In this dissertation I accomplish a comprehensive study on urgent event diffusion through Twitter using the retweet network statistics. My main findings are:
1. With proper choice of category and time window size, the network statistics, especially the dynamic ones can reveal detailed global level and node level diffusion patterns. The results from hashtag K-means clustering and event hashtag binary classification also show these statistics can be helpful to distinguish different types of hashtags.

2. For both the user and hashtags prediction tasks, the network statistics especially the dynamic ones can provide significant improvement over baseline features. The extent of improvement will depend on the experiment settings—including prediction types, choice of training and testing sets, values of prediction parameters etc. The effectiveness of these statistics indicates there is strong past-to-future behavior dependency for both user and hashtag.

3. In both the user and hashtag prediction tasks similar patterns are observed for response distribution and prediction results. Moreover, GLM method outperforms the other two nonlinear methods and proves to be the best prediction method for both tasks on several aspects—results, reliability, interpretability etc. All these may imply

there is strong linear dependency and no significant nonlinear dependency between selected features and response on the log scale, and this dependency may persist regardless of the scopes of prediction—whether on tweet, user or hashtag level.

4. For LDA topic popularity prediction the improvement of network statistics is not significant. I believe the main reason is that the distribution of LDA topic popularity is much more homogenous than the user and hashtags cases. The GLM method does not perform best either, which could be due to the inherent drawback of a linear model in the case of limited sample size. The most reliable approach for this task will be to use the average scores of the other two nonlinear methods with a combination of training sets.

With the above discoveries, my work demonstrates the following contributions:
1. I have developed a comprehensive study of urgent event diffusion on Twitter including information diffusion tracking, user influence and topic popularity prediction. Though there have been studies about urgent events, they do not generally cover the scope that I do. I believe my work can serve as a foundation for further research on urgent event study, which can have large potential social and business impact.

2. I constructed a new early-time prediction (recommendation) framework for both user influence and topic popularity. Given the nature of urgent event, prediction in the early stage of diffusion is a must for useful implementation in practice. I managed to solve the limited information problem, which is faced by every early time prediction approach, by using past event data as training sets for future event data sets prediction. The framework not only produces satisfactory yet consistent results, but is well scalable as well. It best utilizes the information provided by Twitter streaming API and can easily adapt to the "online" prediction mode which maintains continuous input and output streams. Moreover, diffusion tracking or monitoring can be embedded as an add-on function for exploratory analysis. Therefore, with proper additional engineering effort, my framework can provide both diffusion tracking (e.g. summary statistics and visualization) and diffusion prediction (e.g. user prediction and topic prediction) in an online mode by taking streaming input and generating streaming output.

3. I developed innovative network statistics derived from the retweet network for both the monitoring and prediction study. These statistics, especially the dynamic network statistics including rich temporal information about the diffusion, have demonstrated great value in revealing detailed diffusion patterns, predicting user influence and predicting topic popularity. This shows the temporal properties underlying diffusion network can provide much insight on the diffusion study, which has not been stressed in previous study.

4. I obtained robust prediction results from various experiment settings. The improvement from the network statistics is consistent and GLM method gives the best on average performance. This reveals common behavior patterns during the urgent

event diffusion among different levels of entities—tweet, user and topic, which should be a natural but still meaningful conclusion.

All in all, the urgent event diffusion exhibits distinct properties over traditional information diffusion study; therefore the study of urgent event diffusion also requires specific methodology and framework. This is a relatively new research area that only draws limited attention but could have a great potential impact and deserve further exploration. In my dissertation I have established a comprehensive and insightful framework to describe urgent event diffusion on Twitter. I believe my work can be a basis or a good reference for future related study in this area.

5.2 Limitations and expectations

Given the above discoveries and contributions, there are also following limitations that need to be resolved for future work:

1. More data sets are needed to further consolidate all my conclusions. For the hashtag clustering and classification study in Chapter 2, the sample size is less than 50 and this prevents me from reaching a reliable result. Moreover, a well-defined external label is needed to provide a solid validation for hashtag clustering and classification. For the prediction part, I need more data sets from categories of urgent events other than natural disasters—such as breaking news or other explosive events to further validate my findings, although I believe consistent results will be obtained. To go one step further, I will also need data from other platforms—such as Facebook, Instagram or LinkedIn to see if my conclusion is generalizable to use cases other than Twitter.

2. Although I have shown that the network statistics derived from the retweet network are effective in revealing detailed diffusion patterns and predicting future influence (users or topics), there still lacks theoretical foundation to explain these phenomena. A theoretical or empirical guide is needed to figure out the optimal experiment settings in advance—e.g. the optimal time window size for the tracking study, the optimal starting time for the feature collection period (Sandy set vs. Sandy 4.5 set), and the optimal top users or hashtags included (the k value) in the prediction study. This is crucial in my framework since there are many different experimental settings that have to be determined beforehand, especially for the prediction task. In my analysis I choose a series of "standard" settings by exploratory analysis of data and a simple grid search, but for practical implementation it will be much more efficient if there is certain criterion that can be used to make the choice automatically. More specifically, the need for a generally applicable standard is urgent and necessary for the following two tasks: selection of training set and selection of feature collection period starting time. In my study I use KS statistics as the empirical measure for the former task, although it proves to be relatively effective, it still cannot be a reliable and qualified standard. For the later task, the currently available approach is to match

the evolution patterns of training and testing sets, which is far from satisfactory and cannot be used in real-time prediction. Moreover, I have found that different feature sets (baseline features and network features) are good at predicting users or hashtags of different types, thus it would be beneficial to figure out what individual properties (past activity or habit etc.) are leading to the differences. Then I can incorporate these factors into the prediction task to achieve individual automatic feature selection for the best prediction performance.

3. My prediction framework only gives meaningful prediction results for data points that exhibit certain activity during the feature collection period. This is natural since no prediction framework can provide meaningful output given null input. Also my framework is not able to effectively capture data points that experience a rapid response increase, this is because none of the included features can do this well. Generally speaking, limited sources of data should be the one to blame for all the above shortcomings. Given useful external sources, with a slight modification my framework can easily combine features from those sources and overcome the limitations.

4. Currently my framework adopts a wide range of programming libraries across several open source languages. For efficient practical implementation consideration, more engineering effort is needed to merge all of these pipelines into one homogeneous package. Given more time and effort, I would like to write a composite package to achieve all the functionality and build a simple user interface for easy implementation. Ideally, my framework should be able to perform both the tracking and prediction tasks with an "online" mode—taking continuous data input streams, processing them with use specified commands, then generating continuous output streams.

In conclusion, the main direction for the future work would be the cross validation and practical implementation of my whole monitoring and prediction framework. Ideally my framework should be able to help the stake holders who are interested in the diffusion process to either find interesting diffusion patterns or obtain a recommended list of important users or topics, meanwhile with all results continuously updated by feeding new data. While devoting myself to this goal, I would also like to warmly welcome any other interested researchers to join with me to accomplish this project together.

Bibliography

Babyak, M. A. (2004). hat you see may not be what you get: a brief, nontechnical introduction to overfitting in regression-type models. *Psychosomatic medicine*, 411-421.

Bakshy, E. a. (2011). Everyone's an influencer: quantifying influence on twitter. *In Proceedings of the fourth ACM international conference on Web search and data mining*, (pp. 65-74).

Benzi, M. a. (2013). Total communicability as a centrality measure . *Journal of Complex Networks* , 124-149.

Bikhchandani, S. a. (1992). A theory of fads, fashion, custom, and cultural change as informational cascades. *Journal of political Economy* , 992-1026.

Blei, D. M. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 993-1022.

Bollen, J. a. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 1-8.

Borge-Holthoefer, J. a.-B. (2013). Cascading behaviour in complex socio-technical networks. *Journal of Complex Networks*, 3-24.

Borondo, J. a. (2012). Characterizing and modeling an electoral campaign in the context of Twitter: 2011 Spanish Presidential election as a case study. *Chaos: an interdisciplinary journal of nonlinear science*, 22.

Castillo, C. M. (2011). Information credibility on twitter. *In Proceedings of the 20th international conference on World wide web*, (pp. 675-684).

Cataldi, M. D. (2010). Emerging topic detection on twitter based on temporal and social terms evaluation. *In Proceedings of the Tenth International Workshop on Multimedia Data Mining*, (p. 4).

Cha, M. H. (2010). Measuring user influence in twitter: The million follower fallacy. *ICWSM*, (pp. 10-17).

Chen, W. W. (2010). Scalable influence maximization for prevalent viral marketing in large-scale social networks. *In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 1029-1038).

Cheng, J. a. (2014). Can cascades be predicted? *Proceedings of the 23rd international conference on World wide web*, (pp. 925-936).

Costa, J. S. (2013). Defining semantic meta-hashtags for twitter classification. In International Conference on Adaptive and Natural Computing Algorithms. *In International Conference on Adaptive and Natural Computing Algorithms*, (pp. 226-235).

Evans, D. (2010). *Social media marketing: the next generation of business engagement.* John Wiley & Sons.

Galuba, W. a. (2010). Outtweeting the twitterers-predicting information cascades in microblogs. *Proceedings of the 3rd conference on Online social networks.*

Godin, F. S. (2013). Using topic models for twitter hashtag recommendation. *In Proceedings of the 22nd International Conference on World Wide Web*, (pp. 593-596).

Grier, C. T. (2010). @ spam: the underground on 140 characters or less. *In Proceedings of the 17th ACM conference on Computer and communications security*, (pp. 27-37).

Gruhl, D. G.-N. (2004). Information diffusion through blogspace. *In Proceedings of the 13th international conference on World Wide Web*, (pp. 491-501).

Hong, L. a. (2010). Empirical study of topic modeling in twitter. *In Proceedings of the first workshop on social media analytics*, (pp. 80-88).

Hong, L. a. (2011). Predicting popular messages in twitter. *Proceedings of the 20th international conference companion on World wide web*, (pp. 57-58).

Hui, C. a.-I. (2012). Information cascades in social media in response to a crisis: a preliminary model and a case study. *Proceedings of the 21st international conference companion on World Wide Web* , (pp. 653-656).

Jenders, M. a. (2013). Analyzing and predicting viral tweets. *Proceedings of the 22nd international conference on World Wide Web companion*, (pp. 657-664).

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 604-632.

Kupavskii, A. a. (2012). Prediction of retweet cascade size over time. *Proceedings of the 21st ACM international conference on Information and knowledge management*, (pp. 2335-2338).

Kwak, H. L. (2010). What is Twitter, a social network or a news media? *In Proceedings of the 19th international conference on World wide web. ACM.*, (pp. 591-600).

Laflin, P. a. (2013). Discovering and validating influence in a dynamic online social network. *Social Network Analysis and Mining*, 1311-1323.

Lambrecht, K. (1996). *Information structure and sentence form: Topic, focus, and the mental representations of discourse referents.* Cambridge university press.

LeCun, Y. B. (2015). Deep learning. *Nature*, 436-444.

Lerman, K. a. (2010). Information contagion: An empirical study of the spread of news on Digg and Twitter social networks. *ICWSM* , (pp. 90-97).

Livne, A. a. (2011). The Party Is Over Here: Structure and Content in the 2010 Election. *ICWSM*, (pp. 17-21).

Ma, Z. a. (2013). On predicting the popularity of newly emerging hashtags in twitter. *Journal of the American Society for Information Science and Technology*, 1399-1410.

Mantzaris, A. V. (2013). Dynamic network centrality summarizes learning in the human brain . *Journal of Complex Networks*, 83-92.

Mathioudakis, M. a. (2010). Twittermonitor: trend detection over the twitter stream. *In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, (pp. 1155-1158).

Mehrotra, R. S. (2013). Improving lda topic models for microblogs via tweet pooling and automatic labeling. *In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, (pp. 889-892).

Merton, R. K. (1968). The Matthew effect in science. *Science*, 56-63.

Mimno, D. W. (2011). Optimizing semantic coherence in topic models. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, (pp. 262-272).

Morales, A. J. (2014). Efficiency of human activity on information spreading on Twitter. *Social Networks*, 1-11.

Ozdikis, O. S. (2012). Semantic expansion of hashtags for enhanced event detection in Twitter. *In Proceedings of the 1st international workshop on online social systems.*

Petrovic, S. a. (2011). RT to Win! Predicting Message Propagation in Twitter . *ICWSM.*

Porteous, I. N. (2008). Fast collapsed gibbs sampling for latent dirichlet allocation. *In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 569-577).

Ramage, D. D. (2010). Characterizing microblogs with topic models. *ICWSM*, (p. 1).

Rattanaritnont, G. a. (2012). Analyzing patterns of information cascades based on users' influence and posting behaviors. *Proceedings of the 2nd Temporal Web Analytics Workshop* , (pp. 1-8).

Romero, D. M. (2011). Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. *Proceedings of the 20th international conference on World wide web*, (pp. 695-704).

Simmie, D. V. (2014). Ranking twitter influence by combining network centrality and influence observables in an evolutionary model. *Journal of Complex Networks*, 495-517.

Stieglitz, S. &.-X. (2013). Emotions and information diffusion in social media—sentiment of microblogs and sharing behavior. *Journal of Management Information Systems*, 217-248.

Stringhini, G. K. (2010). Detecting spammers on social networks. *In Proceedings of the 26th annual computer security applications conference* , (pp. 1-9).

Swaroop, P. J. (2014). *Influence in Microblogs: Impact of User Behavior on Diffusion and Engagement.*

Taxidou, I. a. (2014). Online analysis of information diffusion in twitter. *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, (pp. 1313-1318).

Thomas, K. G. (2011). Suspended accounts in retrospect: an analysis of twitter spam. *In Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, (pp. 243-258).

Tsur, O. a. (2012). What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities. *In Proceedings of the fifth ACM international conference on Web search and data mining*, (pp. 643-652).

Vicient, C. a. (2014). Unsupervised semantic clustering of Twitter hashtags. *In Proceedings of the Twenty-first European Conference on Artificial Intelligence*, (pp. 1119-1120).

Wu, S. a. (2011). Who says what to whom on twitter. *Proceedings of the 20th international conference on World wide web*, (pp. 705-714).

Yang, S. H. (2014). Large-scale high-precision topic modeling on twitter. *In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 1907-1916).

Zhou, Z. a. (2011). Information resonance on Twitter: watching Iran. *Proceedings of the first workshop on social media analytics*, (pp. 123-131).