ABSTRACT

Title of dissertation: PREDICTION IN SOCIAL MEDIA FOR MONITORING AND RECOMMENDATION

Shanchan Wu, Doctor of Philosophy, 2012

Dissertation directed by: Professor Louiqa Raschid
Department of Computer Science

Social media including blogs and microblogs provide a rich window into user online activity. Monitoring social media datasets can be expensive due to the scale and inherent noise in such data streams. Monitoring and prediction can provide significant benefit for many applications including brand monitoring and making recommendations. Consider a focal topic and posts on multiple blog channels on this topic. Being able to target a few potentially influential blog channels which will contain relevant posts is valuable. Once these channels have been identified, a user can proactively join the conversation themselves to encourage positive word-of-mouth and to mitigate negative word-of-mouth.

Links between different blog channels, and retweets and mentions between different microblog users, are a proxy of information flow and influence. When trying to monitor where information will flow and who will be influenced by a focal user, it is valuable to predict future links, retweets and mentions. Predictions of users who will post on a focal topic or who will be influenced by a focal user can yield valuable recommendations.

In this thesis we address the problem of prediction in social media to select social media channels for monitoring and recommendation. Our analysis focuses on individual authors and linkers. We address a series of prediction problems including future author prediction problem and future link prediction problem in the blogosphere, as well as prediction in microblogs such as twitter.

For the future author prediction in the blogosphere, where there are network properties and content properties, we develop prediction methods inspired by information retrieval approaches that use historical posts in the blog channel for prediction. We also train a ranking support vector machine (SVM) to solve the problem, considering both network properties and content properties. We identify a number of features which have impact on prediction accuracy. For the future link prediction in the blogosphere, we compare multiple link prediction methods, and show that our proposed solution which combines the network properties of the blog with content properties does better than methods which examine network properties or content properties in isolation. Most of the previous work has only looked at either one or the other. For the prediction in microblogs, where there are follower network, retweet network, and mention network, we propose a prediction model to utilize the hybrid network for prediction. In this model, we define a potential function that reflects the likelihood of a candidate user having a specific type of link to a focal user in the future and identify an optimization problem by the principle of maximum likelihood to determine the parameters in the model. We propose different approximate approaches based on the prediction model. Our approaches are demonstrated to outperform the baseline methods which only consider one network

or utilize hybrid networks in a naive way. The prediction model can be applied to other similar problems where hybrid networks exist.

PREDICTION IN SOCIAL MEDIA FOR MONITORING AND RECOMMENDATION

by

Shanchan Wu

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
2012

Advisory Committee:
Professor Louiqa Raschid, Chair/Advisor
Professor Amol Deshpande
Professor Alan Sussman
Professor William Rand
Professor Richard J. La

Acknowledgments

First, I would like to thank my advisor Professor Louiqa Raschid for her advice and support. This thesis would not have been possible without her great help. I thank her for spending a lot of time in this thesis work, with a lot of discussions, tremendous editing work and constructive suggestions.

I would also like to thank all my other coauthors, who have made important contributions to this thesis. Professor Louiqa Raschid and Professor William Rand have been especially influential and important in shaping the work here. Dr. Tamer M. Elsayed has also been a pleasure to work with and I appreciate his contributions to Chapter 3. I would like to thank Dr. Derek Monner for his help in data collection. I would also like to thank Professor Yogesh V. Joshi and Professor Samir Khuller for their comments.

I would like to thank Professor Amol Deshpande, Professor Alan Sussman, and Professor Richard J. La for spending their precious time to serve on my thesis committee, and for their advice and comments.

Finally, I'd like to thank my parents for always being supportive, and my wife, Fang Wang, who is always there with me, for her support and love and understanding.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

We address the problem of prediction in social media to select social media channels for monitoring and recommendation. Our analysis focuses on individual authors and linkers. The general question we are trying to answer is: given a focal post or a focal user, which other actors in social media will carry out relevant and interesting actions in the near future? We address a series of prediction problems including the future author prediction problem and the future link prediction problem in the blogosphere, as well as prediction in microblogs such as twitter.

Given data streams representing blog posts on multiple blog channels and a focal query post on some topic of interest, one of our objectives is to predict which of those channels are most likely to contain a future post that is relevant, or similar, to the focal query post. We denote this task as the future author prediction problem (FAP). This problem has applications in information diffusion for brand monitoring and blog channel personalization and recommendation.

An essential element of social media, particularly blogs, is the hyperlink graph that connects various pieces of content. There are two types of links within the blogosphere; one from blog post to blog post, and another from blog post to blog channel (an event stream of blog posts). These links can be viewed as a proxy for the flow of information between blog channels and to reflect influence. Given this assumption about links, the ability to predict future links can facilitate the monitoring of information diffusion, making recommendations, and word-of-mouth (WOM) marketing. In one part of thesis we address the future link prediction in the blogosphere.

Research in diffusion and influence, or contagion models, typically assume that the network is homogeneous; this can simplify the model that is used. Microblogs such as twitter are an exemplar of a hybrid network. There is a network of followers. In addition, there is an implicit network of users who retweet other users, and users who mention other users. Such retweets and mentions are an important proxy for influence and must be considered by any model. We choose a focal user and then try to predict those users who will retweet a focal user's tweets, and/or who will mention a focal user, in the near future. We propose a prediction model and different approaches to utilize the hybrid network for prediction.

These predictions in social media will provide practical recommendations. For example, based on these predictions, brand managers could actively participate in social media conversations, and potentially contact authors proactively, to provide them with accurate information or to address any concerns. We also study the recommendations from the following perspectives: novel linkers and authors; authors with diverse profiles; authors who will both write and link to the focal topic; and how the characteristics of focal users such as the size of the follower network, or the level of sentiment averaged over all tweets would impact on the quality of personalized recommendations; and the centrality of the recommended users.

1.1 Motivation and Applications of Future Author Prediction

Social media such as the blogosphere has emerged as an important source of online activity. Social media creates new online content through a form of crowd-sourcing; this "wisdom of the masses" approach facilitates the creation of content that is both timely and diverse, but, this method also makes systematic examination difficult since the content is constantly changing and may be dominated by noise and irrelevant posts. Users who rely on the blogosphere to keep track of events or trends, or to follow a conversational thread between several contributing participants, have to face the daunting task of keeping up to date with potentially thousands of blog channels and their posts, and filtering out relevant content.

In the blogosphere, a blog channel is a stream of posts (blog entries) originating from a single author or source (i.e., a blogger), or a group of bloggers. It is typically visualized as a web page from which a collection of posts can be accessed. Figure 1.1 illustrates 4 blog channels. Posts p_1, p_7 and p_{10} in blog channel b_1 and posts p_4, p_6 and p_8 in blog channel b_2 all represent a conversation on one topic, while posts p_3

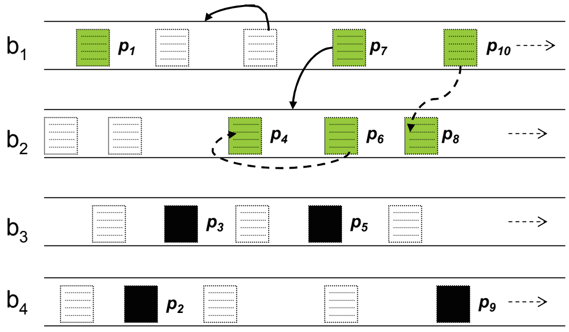


Figure 1.1: Blog channels.

and p_5 in blog channel b_3 and p_2 and p_9 in blog channel b_4 refer to a different topic. Given a query post p_1 on blog channel b_1, our goal is to predict that blog channels b_1 and b_2 are high value blog channels that will contain similar posts in the future, before we actually observe these future posts.

Understanding these conversations and how they diffuse through social media can have ramifications for companies involved in brand monitoring [63]. Word-of-mouth has always played a significant role in information propagation about brands among consumers [10], but until the advent of social media it has been difficult to track these discussions. However, the great opportunity of social media is also a problem; social media generates so much data that monitoring of brand conversations can be very difficult. Being able to target even a few highly relevant blog channels and their most important posts is an advantage because of the impact social media has on consumer decisions [23]. A company that can identify highly relevant blog channels and topics can use this information to either diffuse explosive situations (e.g., Gap logo fiasco in the Fall of 2010), or to enhance positive brand experiences (e.g., the adoption of penny loafers by clubbers in NYC). Once a potentially important channel has been identified, a company can then join the conversation themselves [37] to encourage positive word-of-mouth.

To enable this to happen, it is necessary to develop tools for marketing managers that identify which blog channel is likely to discuss Topic X as it relates to their brand, or which bloggers will respond to Topic X. It allows the manager to devote resources to that particular blog channel, and potentially even reach out to the blogger proactively. Moreover, if the manager can make a prediction about how many bloggers are going to blog about Topic X, how frequently they will post, or for how long they will continue to post about Topic X, then they can determine if the company needs to craft a response or if the topic will simply die out on its own. This allows the manager to make more educated decisions as to how many resources to devote to monitoring. These predictions are also recommendations as to which blog channels should be monitored for a particular topic. Note that recommendations could also be used to allocate scarce time among the large number of blogs channels, to identify those channels of highest relevance.

To formalize these questions, we pose the following problem: *Given a specific query post on some topic on a blog channel, what other blog channels are likely to post on the same topic in the (near) future?* The term *query post* refers to a post that will be used for search and for comparison [1] This task of predicting the author(s) of future posts, which we call the Future Author Prediction Problem (FAP), is difficult. First, a good solution must predict the content of future posts to determine if they will be relevant to the query post. Then, for the relevant posts, one must predict the author blog channels. Further, the joint expectation for these two prediction tasks must be maximized and the Top K must be chosen. We note that predicting the content of a future post is difficult since there are few features that can be used for prediction. On the other hand, predicting the author of a future post is somewhat easier since we can consider the historical posts in a blog channel to build a profile of the author, and instead of predicting the content of a post, we are simply predicting

[1] The term *query document* was also used by Yang et al [113] to refer to a document whose phrases are used as queries.

who will post on a particular topic.

Solving this prediction problem will provide a costs savings to those interested in brand monitoring and recommendations since we can avoid the considerable expense of monitoring and tracking all future posts on all blog channels. We can also reduce the number of times we need to carry out the expensive tasks of data cleaning, extraction and analysis of posts by recommending that only high value blog channels be monitored. We develop prediction methods inspired by (naive) information retrieval approaches that use historical posts in the blog channel for prediction. We also train a ranking support vector machine (SVM) to solve the problem. We identify a number of features which have impact on prediction accuracy and can potentially be used to indicate a confidence level of a prediction.

1.2 Motivation and Applications of Future Link Prediction

Links between different blog channels typically indicate the direction of information flow in the blogosphere. While analyzing the structure of the links can help understand the propagation of information, information diffusion and influence have also been extensively studied as a marketing strategy. A typical objective is to understand how diffusion will impact the decision of adopting new products. Diffusion models [10] examine how influence propagates in the network [32, 53, 83]. When trying to monitor where information will flow and who will be influenced, it is important to predict future links since these links are a proxy for future spread of influence.

Predicting future links is also useful when recommending interesting blog channels to readers. A link from a post to another could indicate that the blogger is following the topic(s) that is discussed in the post to which a link is placed. Similarly, a link from a post to a blog channel could indicate that the blogger is a fan of the author of the blog channel and is influenced by this author. Since recently published posts typically attract more readers than older posts, future link prediction for recent posts is of greater significance. In addition, there has not been enough time for recent posts to build a following or to have many links or references from other posts, i.e. it may take some hours or days for links to be created. Hence, future link prediction is an important objective when making recommendations for recent posts.

Finally, the influence of a blogger is invaluable as a marketing strategy. Word-of-mouth (WOM) marketing, which refers to the passing of information from person to person and includes blog and other types of social media, is believed to increase the credibility of product information. Research points out that individuals are more inclined to believe WOM marketing than more formal forms of promotion methods [39]; the receivers of WOM referrals tend to believe that the communicator is speaking honestly. Hence it is potentially a promising way to utilize social media for advertisement.

For WOM marketing to be a success, one has to identify influential blog channels. The two important factors which affect the influence of a blog channel are the content posted on the blog channel and the links pointing to the blog channel or to posts in the blog channel. We note that while links help increase the influence of the blog channel, links also serve a more interesting function since the presence of links could indicate the degree of influence of the target blog channel. Further, both the number of links to the target blog channel as well as the source of those links can indicate the influence of the target blog channel. A link from an authoritative or influential blog channel can indicate greater influence than a link from a less influential blog channel. Future link prediction can thus be a key element of successful WOM marketing strategy.

To summarize, links between blog channels are a proxy for information flow and influence. They are useful for both making recommendations as well as a measure of influence or authority for WOM marketing. Further, recent posts are very interesting to readers, but their recency also means that they may not have attracted many links. All of these factors motivate the importance of the problem of future link prediction.

Link prediction has been studied in social networks, relational datasets, labeled entity-relationship graphs, etc. Two classes of approaches have been successfully applied to this problem. One class focuses on topological features of graphs [64, 91]. The second class uses robust machine learning approaches such as spectral transformation [57], the heat diffusion kernel [50], Markov Random Field Model [103], collective classification [100], etc.

Taskar et al. [100] applied a collective classification approach to predict links in relational data and entity-relationship graphs. This approach works well for labeled graph datasets where there are strong relationships (e.g., an advisor-advisee relationship), and/or the nodes have rich feature labels (e.g., the nodes are labeled as faculty, student, staff and so on). We do not expect such methods to perform well in the blogosphere since there are no strong relationship types nor are there rich feature labels. A blog post is represented by a bag of words. While there are techniques to extract named entities from a bag of words, we cannot always expect to obtain a rich set of feature labels for the blogosphere. Another limitation is that such classification approaches may not scale well to the large graphs typical of social media. Taskar at al. evaluated their methods on a dataset of less than 3 thousand webpages; our dataset, includes over 42 thousand blog channels with more than 2 million blog posts.

An array of methods for link prediction based on topological features of graph were studied by Liben-Nowell and Kleinberg [64] who evaluated them on co-authorship networks. We know that co-authorship networks are dense networks while the network of the blogoshpere is a sparse network. Thus, methods which work well for dense networks may not work well for sparse networks. We have implemented several link based prediction methods based on suggested techniques and metrics from [64].

Informally, the future link prediction problem in the blogoshpere is as follows: Given a target blog channel, our objective is to predict the Top K blog channels that will contain links to the target blog channel in the near future. While the prediction may use historical links, we note that these may be sparse. Further, recall that a blog channel is represented by a stream of posts; thus, the underlying graph dataset is composed of posts within a blog channel. Future link prediction in this context may also benefit from aggregating all the historical posts to maintain a profile for

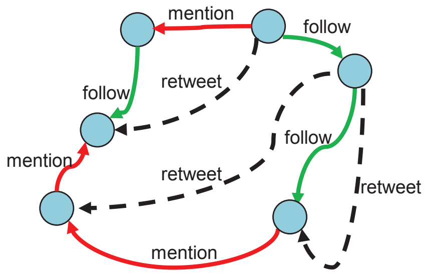


Figure 1.2: Microblog networks.

the blog channel. We demonstrate it with a method by combining the network properties and the content properties.

1.3 Motivation and Applications of Prediction with Microblogs

Unlike blogs, where the network created by the links between different blog channels is homogeneous, in microblogs, there exist hybrid networks. On a microblog site such as Twitter, one can follow a user and read tweets or search for tweets based on queries. One can initiate a new conversation by tweeting or one can interact by mentioning a user. One can also participate in the diffusion of a topic by retweeting. All these interactions create a dynamic and rich social network. There is so rich information including hashtags and URLs in a tweet even though the length of each tweet is limited. For example, there is a limitation of 140 characters for each tweet in Twitter. Figure 1.2 shows the different networks in a microblog. Analysis and prediction in microblogs would have some different properties and different challenges. We are interested in analyzing the influence in microblogs such as twitter from the individual perspective. We want to understand who will be influenced by a particular focal user in the microblog. Specifically, in microblogs, retweets and mentions are a proxy of being influenced [21]. To understand who will be influenced in the future, we focus on predicting who will retweet a focal user's tweets, and who will mention a focal user.

Most of the existent influence models assume that one can only influence her neighbors, for example, the Linear Threshold Model and Independent Cascade Model [38, 53]. The definition of neighbors is that there are edges in the network between the users. The edges are concrete and observable. For example, in a friendship relationship network, one can only influence her friends. In a disease spread network, disease can only spread to the people who are her neighbors and have direct contact.

Our observation for microblogs is that the neighborhood does not completely identify the area of influence. For example, in our experimental dataset (described in Section 5.4), more than 40% of the mentions are from outside the follower network. Influence can spread outside of the follower network in microblogs. One can retweet any users' tweets or mention any users who may not be her friends.

Twitter as an example of a microblog can be composed of multiple networks. Based on the follower relationship in microblogs, follower network can be constructed. Retweet actions and mention actions can add some linkages between users and hence retweet network and mention network can also be constructed.

We can formalize the retweet prediction and mention prediction to be a link prediction problem. Unlike traditional link prediction [64], where the network is usually homogeneous, here we have an evolving hybrid network. We would expect that exploiting the hybrid networks for prediction would help improve the prediction accuracy. We propose a general prediction model to utilize the hybrid networks for the prediction and propose two approximate approaches based on the prediction model.

1.4 Contributions

In this thesis, we study the prediction problems in social media, including blogosphere and microblogs. For both types of social media, we define the prediction problems, analyze the problem properties and features, present multiple solutions, conduct analysis of the prediction precision and exploit their applications.

The prediction tasks in social media are challenging. The social media data is usually very large. Solutions applicable to a small dataset might not be applicable to the huge social media dataset. There is usually noisy data. Effort must also put into preprocessing and feature selection. Further more, the social media is dynamic. Networks and content evolve, which will make the prediction tasks more challenging.

The prediction accuracy is important for high quality of recommendations. Higher prediction accuracy indicates a higher confidence level of recommendation, and would usually be more helpful for the users (e.g. brand managers) to make right actions. For predicting novel authors who have not talked about the focal topic in the history but will talk about it in future, and the novel linkers who have not linked to the focal user in the history but will link to the focal user in the future, although it would be more difficult and the prediction accuracy would be lower, the values for predicting those users are very high and the recommendations based on it would still be very helpful. This thesis work would help to understand how much accuracy could be achieved for the prediction tasks in social media, and what are the important features for different prediction tasks.

For the prediction in the blogoshpere, we focus on two prediction problems. One is the future author prediction problem (*FAP*), which is to answer who will post on a focal topic. The other is the future link prediction problem (*FLP*), which is to answer who will link to a focal blog channel.

In the blogosphere, there are rich content features. We build a profile based on bag of words for each blog channel. We consider document similarity between the profiles of different blog channels and between different blog posts and between blog channel profiles and blog posts as content features. We also consider several metrics based on the link structure within the blogophere as network features. In addition, we consider named-entities and external links which point to the outside of the blogoshpere as extra features. Based on these features, we propose multiple prediction methods, and make extensive evaluations.

For the prediction in the Blogosphere, our contributions are as follows:

- We define the future author prediction problem (*FAP*) and the future link prediction problem (*FLP*), develop multiple solutions for the problems, and perform extensive evaluations on a large social media dataset.
- We train a ranking SVM to utilize multiple features to improve the prediction accuracy.
- For the future author prediction problem (*FAP*), we identify several blog chan-

nel properties which have impact on prediction accuracy, including diffusion stage and blog channel consistency etc. Those properties can potentially be used to indicate a confidence level of a prediction.

- For future link prediction problem (*FLP*), we compare multiple link prediction methods, and demonstrate that a method which combines the network properties of the blog with content properties does better than methods which examine network properties or content properties in isolation.

We formalize the retweet prediction and mention prediction problem in microblogs to be a link prediction problem. Unlike the traditional link prediction, where the network is usually homogeneous, here we have evolving hybrid networks, which include retweet network, mention network, and follower network. We would expect that exploiting the hybrid networks for prediction would help improve the prediction accuracy. In this part of thesis work, we study how to utilize the hybrid network to improve the prediction accuracy.

For the prediction in microblogs, our contributions are as follows:

- We define a challenging link prediction problem for an evolving hybrid network. We propose a prediction model to utilize the hybrid network for prediction. In this model, we define a potential function that reflects the likelihood of a candidate user having a specific type of link to a focal user in the future and identify an optimization problem by the principle of maximum likelihood to determine the parameters in the model.

- We propose different approximate approaches, *WT-COM-BON* and *MIX-PATH*, based on the prediction model. We perform an extensive evaluation over a microblog network and a stream of tweets from Twitter. Our approaches are demonstrated to outperform the baseline methods which only consider one network or utilize hybrid networks in a naive way. The prediction model can be applied to other similar problems where hybrid networks exist.

- We consider a subset of retweets and mentions from novel users, i.e., they do not retweet or mention the focal user in history. Our approaches show significant improvement over baseline methods for this challenging problem.

1.5 Outline

Chapter 2 discusses some related work. In Chapter 3, we address the future author prediction problem. We define the problem and investigate the properties of the problem. We propose our solutions and present our experimental results. In Chapter 4, we address the future link prediction problem in the social media context. We identify a combination of content and network based features and train a ranking SVM to use these hybrid features. We compare different methods and show our experimental results. In Chapter 5, we address the prediction in social media where hybrid networks exist. We propose a general prediction model as an optimization problem to utilize the hybrid networks for the prediction and propose two approximate approaches based on the prediction model. We conduct an extensive evaluation on a microblog network and a stream of tweets from Twitter. In Chapter 6, we study two recommendation cases based on our prediction work in social media. Finally Chapter 7 concludes the thesis.

Chapter 2

Related Work

2.1 Related Work on Future Author Prediction

Blog channel tracking and online news monitoring have become topics of research interest recently. For instance, the dynamics of the news cycle has been studied through the tracking of topics and memes (represented by soundbites) as they disseminate and evolve over time [60]. On the blog side, blogTrust [102] examined the sudden convergence of communities of bloggers and their connection to real world events, while El-Arini et. al. [33] provided efficient techniques to sample posts in the blogosphere for personalized coverage and ranking. Since most of this work focused on tracking information as it spreads across communication channels, our high value blog channel prediction can complement this work by prioritizing which channels to monitor to achieve a better use of scarce resources.

Our work could also be beneficial even when the goal is a full catalog of all blogs. For instance, BlogScope [8] has been very successful at online analysis of high volumes of blog channels; at present it indexes over 39 million blog channels and almost a trillion posts and updates the indexes every three hours [8]. Continuously updating an inverted index, can incur significant overhead, and so our blog channel prediction could provide a significant benefit by prioritizing updates to the index, based on user interests.

Another area of related work is topic or event detection and topic tracking (TDT) [6] , which is well studied in many domains including news [112], and there are several excellent methods that address the challenge of TDT [6, 68]; some of which examine TDT within the context of social media [11, 65, 81, 89, 105]. However, all of this work addresses the problem of identifying and tracking topics in an already extant corpus, as opposed to predicting a channel to monitor for a future post on a given topic, which is the focus of this research.

One of our solutions is based on the ranking SVM [51]. The ranking SVM is a learning-to-rank method and there are some other learning-to-rank methods in the literature, such as RankBoost [35], RankNet [16], AdaRank [111] and BayesRank [58].

2.2 Related Work on Future Link Prediction

Link prediction is a challenging problem and has been studied in social networks, relational datasets, labeled entity-relationship graphs, etc. Several approaches have been successfully applied to this problem. One class of solutions focuses on topological features of graphs [64, 91]. A second class uses robust machine learning approaches such as spectral transformation [57], the heat diffusion kernel [50], Markov Random Field Model [103], collective classification [100], etc. An excellent summary and some models are presented in [59].

Taskar et al. [100] applied a collective classification approach to predict links in relational data and entity-relationship graphs. This approach works well for labeled graph datasets where there are strong relationships (e.g., an advisor-advisee relationship), and/or the nodes have rich feature labels that can be uniformly applied (e.g., the nodes are labeled as faculty, student, staff and so on). We do not expect such methods to perform well in the blogosphere since there are no strong relationship types nor are there uniform labels. A blog post is essentially a bag of words. While there are techniques to extract named entities from a bag of words, we cannot always expect to obtain a consistent set of labels for blogs. Another limitation is that such classification approaches may not scale well to the large graphs typical of social media. Taskar et al. evaluated their methods on a dataset of less than 3 thousand webpages; our dataset includes over 42 thousand blog channels with more than 2 million blog posts.

An array of methods for link prediction based on topological features were presented in Liben-Nowell and Kleinberg [64] who evaluated them on co-authorship networks. We evaluate some of these methods in the blogosphere and also compare them with other methods based on additional features.

Link prediction has also been studied in different domains [73], such as social network analysis, bioinformatics, and computer network systems analysis. In social network analysis, some work has been done on predicting friendship links [115], email links [76], co-authorship links [76], semantic relationship links such as subordinate-manager [30] and advisor-of [100]. In bioinformatics, some work has been done on predicting the existence of edges representing physical protein-protein interactions [46, 97, 114], and domain-domain interactions [29], and regulatory interactions [4]. In computer network systems, some work has been done on inferring relationships between autonomous systems and service providers [94], as well as inferring unobserved connections between routers.

2.3 Related Work on Prediction with Microblogs

Cha et al. in [21] have studied measuring user influence in a microblog, i.e., Twitter. They found that retweets and mentions are more important for influence rather than the indegrees of the follower network. Predicting the most influential users in a microblog has been addressed in [95]. Other than microblog, in blogosphere, links between different blog channels indicate influence. Future link prediction in the blogosphere has been addressed in [110]. To the best of our knowledge, in microblogs, prediction of retweets and mentions from individual-level has not been addressed before.

Retweet and mention prediction problem can be formalized as a link prediction problem. Link prediction problem has been studied in various applications in social networks, relational datasets, labeled entity-relationship graphs, etc. An array of topological methods for link prediction were studied by Liben-Nowell and Kleinberg [64] who evaluated them on co-authorship networks. Machine learning approaches have also been applied to link prediction, like spectral transformation [57], the heat diffusion kernel [50], Markov Random Field Model [103], collective classification [100], Ranking SVM [110] etc. In most of the previous link prediction work, the situation that there may be different types of links between two nodes has not been considered. Our work focuses on how to use these coexistent different types of links for prediction.

A model based on composite network has been applied to predict mobile application installation [79]. The authors collected different social networks using built-in sensors, including Bluetooth proximity network, call log network, etc. However, unlike our problem, application installation which they tried to predict is not part of the networks. Instead, it can just be looked as one property (installation or not installation) of a node (user) in the composite network. Their prediction model is based on an assumption that whether a user will install an application or not is depended only on his neighbors in the composite network. They solved an optimization problem to create the composite network. Since similar assumption does not hold for our problem, their model could not be directly applied to our problem. Our prediction problem with microblogs can be categorized as link prediction while their problem can be categorized as node property prediction. Further more, their optimization solution would meet the scalability issue.

There are many diffusion and influence models for social networks. For example, the Linear Threshold Model and the Independent Cascade Model have been widely studied [53, 38]. For the Linear Threshold Model, in each step, a user will be activated (influenced) if the total weight of her active neighbors is greater than a threshold. For the Independent Cascade Model, each active user has a single opportunity with some probability to activate each of her inactive neighbors.

These prior models have limitations when applied to microblogs. One limitation is that these prior models are often at the aggregate level, e.g. at the level of a topic [99]. One popular aggregate level influence challenge is the influence maximization problem. It was first formulated as a discrete optimization problem in [53] and was also studied by others [22]. The target of influence maximization is to select an initial set of users who eventually influence the largest number of people in the network. Predicting the degree of influence has also been studied; for example, a regression model is used to predict the influence of a user [7]. However, no previous research models influence at the individual level, e.g., *who will mention user u or retweet user v*?

Another limitation is that both the Linear Threshold Model and the Independent Cascade Model typically assume that one can only influence her immediate neighbors. The definition of neighbors is that there are edges in the network between these users. The edges are concrete and observable. For example, in a friendship relationship network, one can only influence her friends. In a disease spread network, some diseases can only spread through direct contact with a user.

2.4 Related Work on Recommendation

Platforms that aid in recommending relevant blog posts have been developed for a number of commercial websites. For example, Google Blog Search [1], Yahoo! Buzz [2], Digg [31], and Blogpulse [14]. A majority of these websites recommend posts that are handpicked by editors or that are voted on by users. Some websites recommend posts automatically; this is typically based on ranking posts on their global popularity or using other global metrics.

Two common approaches to recommendation are collaborative filtering [26, 66] and content-based filtering [80]. Collaborative filtering aims to learn user preferences and make recommendations by correlating the user's past activity with data from the entire user community. In a content-based approach, documents are recommended to a user if they are similar to documents that the user previously liked, where similarity is based on content. [41] explores a variety of hybrid recommendation strategies including content-based techniques and collaborative filtering, based on the followees and followers of users. There are some models of collaborative filtering, such as matrix factorization [55], Bayesian networks [42], restricted Boltzmann machine [87] and topic models [104]. Existing collaborative filtering algorithms do not distinguish between current and historical data. An online evolutionary approach [67] extends the widely used neighborhood based algorithms by using instance weighting techniques to incorporate temporal information while updating neighborhood similarities.

Although our recommendation approach in the blogosphere exploits content, we are not recommending similar documents or similar posts to users. We are instead recommending blog channels that are likely to be future authors of some focal topic or future linkers to some focal blog channel. We are also interested in novelty, i.e., identifying *Novel Authors* who do not have historical posts or *Novel Linkers* who have not linked to the focal blog channel, and diversity, i.e., identifying authors who write about the focal subject in different contexts.

In our recommendation work in microblogs, we analyze the factors of sentiment and the network centrality. There has been a lot of work looking at Twitter sentiment [9, 12, 27, 56, 75, 78, 101]. Network centrality has also been studied for many applications such as targeted advertisement and recommendation [3], routing protocols [48, 25, 49], content sharing [70], epidemiological modeling [54, 98], network reliability [5, 72], urban planning [82] and resource provisioning [92].

Chapter 3

Future Author Prediction in the Blogosphere

3.1 Introduction

Social media is playing an ever increasing role in the marketing of new products and brands; this is in part because word-of-mouth communication, such as social media, have a dramatic effect on consumers' purchase decisions [23]. Brand managers must pay attention to social media so that they can monitor the pulse of conversations that concern their brand [63]. They can identify emerging discussions and join the conversations, possibly to encourage positive word-of-mouth [37].

Prioritizing or personalizing blogs or other social media channels is essential since managers do not have time to monitor the entire blogosphere. It is also useful to determine how quickly posts on a focal topic will spread across the blogosphere, and more importantly, which bloggers will post on that focal topic in the near future.

As an illustration, consider the Gap logo fiasco in the Fall of 2010. Gap introduced a new logo, changing the iconic logo it had for 20 years almost overnight. There was an immediate outpouring of negative comments about the new logo on Twitter, Facebook, and across the blogosphere; Gap quickly reverted to the old logo. It would have been very helpful if a brand manager at Gap could have detected a blog post on this topic early on, and then predicted whether or not that conversation would spread to other blogs, and which bloggers, if any, would write about the topic. If the brand manager had this information, then she could select which blogs to monitor. She could participate in conversations, or even contact the bloggers ahead of time, to provide more accurate information and to keep them up to speed on the company's response.

To achieve that, it is necessary to develop tools that identify which blog channel is likely to next discuss Topic X (e.g., Gap Logo Redesign) as it relates to a brand (e.g., Gap), or which bloggers will respond to Topic X. To formalize these questions, we pose the following problem: *Given a focal query post on some topic on a blog channel, what other blog channels are likely to post on that topic in the (near) future?* The term *query post* refers to a post that will be used for search and for comparison[1]. We denote this task as the *Future Author Prediction Problem* (*FAP*).

A good solution to the problem must predict the content of future posts to determine if they will be relevant to the query post. Then, for the relevant posts, one must predict the author blog channels. Finally, the joint expectation for these two prediction tasks must be maximized and the Top K authors/channels must be chosen. We note that predicting the content of a future post is difficult since there are few features that can be used for prediction. On the other hand, predicting the author of a future post is somewhat easier since we can consider the historical posts in a blog channel to build a profile of the author.

We consider several solutions to *FAP*. **PROF** and **VOTE** are inspired by information retrieval approaches and exploit historical posts to make a prediction.

[1] A similar term, *query document*, was used by Yang et al [113] to refer to a document whose phrases are used as queries.

We also identify a number of additional features to train a ranking support vector machine for prediction, denoted as **RSVMP**. We test our methods using a blog dataset from Spinn3r [17]. Despite the difficulty of the *FAP* task, all methods provide reasonably accurate results. **PROF** dominates **VOTE** while **RSVMP** dominates both. We also identify multiple characteristics that impact prediction accuracy including diffusion stage (*cRatio*), volume versus author count (V/AC) and blog channel consistency. **RSVMP** can exploit all of these characteristics to improve prediction accuracy.

These characteristics are of great interest since they affect the strategy and efficacy of a brand manager. For instance, if the topic is in the middle of its diffusion across the blogosphere (i.e., a mid-range cRatio), such as halfway through the Gap Logo controversy, then that is a critical period when the brand manager can have the greatest impact on the conversation. Before that time, it may not be clear if the topic will take off, and after that point, the conversation around it slows down, or perhaps has already trended negative. If the brand manager can predict which authors are likely to post in the mid-stage of diffusion, then actions can be taken. Our results show that **RSVMP** achieves accurate predictions under this scenario. It also performs surprisingly well for emerging topics.

Alternately, suppose that the story is not spreading, but is heavily-discussed only by a few authors (i.e, a high V/AC). If these authors are vocal (e.g., have a lot of followers), then it is important to predict new authors; this is another scenario where **RSVMP** can make accurate predictions.

Content-based techniques such as **PROF** are good at predicting the "usual suspects", however, what really concerns a brand manager is when a *difficult-to-predict* blogger or community gets involved. Difficulty increases when bloggers are inconsistent in their posts or because the comments come from a diverse set of bloggers. For instance, in the Gap Logo scenario, the brand manager may typically monitor clothing and fashion blogs, but the controversy may have emerged around blogs of graphic artists. While highly-consistent bloggers are easier to predict, **RSVMP** also performs well in identifying bloggers who are less consistent or have a diversity of profiles.

To summarize, we define a novel and challenging prediction problem *FAP*. We develop multiple prediction methods and complete an extensive experimental evaluation. We show that a ranking SVM can be trained to exploit relevant features and can make accurate and useful predictions for many brand monitoring scenarios. These results are presented in [107].

3.2 Problem Characteristics

3.2.1 Problem Definition

A blog channel is an event stream of posts (blog entries) originating from a single source (a blogger, news agency, organization, etc.). It is typically visualized as a web page from which a collection of posts can be accessed. Figure 1.1 illustrates 4 blog channels. The problem of predicting high value blog channels for monitoring future posts can be defined as follows:

Definition 1 ***Future Author Prediction Problem (FAP):*** *Given a query post*

q posted at time T_q, identify the high value blog channels $B_{q,\Delta T}$ that will contain at least one future post p posted in the interval $(T_q, T_q + \Delta T]$ that is topically similar to q. To make the problem specific, we use a similarity metric M_{sim} and a threshold η. Using this metric, the high value blog channels $B_{q,\Delta T}$ must satisfy the following condition:

$$\forall b \in B_{q,\Delta T} \; \exists \; p \in b \mid T_p \in (T_q, T_q + \Delta T] \wedge M_{sim}(p, q) \geq \eta$$

The goal is to identify up to K author blog channels in $B_{q,\Delta T}$.

The *FAP* is composed of two sub-tasks. The *relevance task (RT)* is to identify unknown future posts p such that $M_{sim}(p, q) \geq \eta$. The second *authoring task (AT)* is to predict the blog channel B_p in which post p appears. The problem is more complex and different from a traditional retrieval problem. For retrieval, the collection of all posts $B_{j,\Delta T}$, for all blog channels j, is known a priori. In contrast, for the *FAP* each future post p and its features are not known. Further, a solution to the *FAP* must maximize the *joint expectation* for both tasks for post p with respect to query q and blog channel B_p, i.e., that the post p is relevant to the query post q, and that B_p is the authoring blog channel for post p.

Since *FAP* is novel and difficult, in order to understand the quality of the results, we will perform an evaluation of the simpler *AT* for a *known* post, i.e., its features are given. While *AT* prediction is simpler, obtaining accurate results may be difficult since there is exactly one authoring blog channel for each post. In comparison, for *FAP*, there may be many authoring blog channels in the ground truth. It should also be noted that *RT* is nearly impossible on its own, since it involves predicting the content of an unknown future post, the *FAP* simplifies this task since a solution to the FAP only needs to identify the blog channel that will post on a similar topic and not the actual content of the post.

3.2.2 Computing Similarity of Posts

3.2.2.1 Similarity Metric

We use the similarity between two posts as a proxy indicating that the two posts are on the same topic. This is both simple and effective. We note that there are sophisticated methods for topic detection, e.g., LDA topic modeling [13] and other topic models such as LSA [28], pLSA [45], LapPLSI [18], LTM [19], DTM [47].

Both the query post as well as matching future posts are represented in the vector space model as a vector of terms. Each element of the vector is a weighted term. Each term is weighted using an information retrieval weighting function. We primarily use the Okapi weighting function [84, 90]. We also use the Okapi similarity metric to determine a similarity score between two posts. A higher term weight means that the corresponding term is more important in that document. A zero term weight is assigned to those terms that do not appear in the document. The following three main factors come into play in the term weight formulation:

- Term Frequency (or tf): Words that repeat multiple times in a document are considered relatively more important.
- Inverse Document Frequency(or idf): Words that appear in many documents

are considered common and relatively unimportant.

- Document Length: When collections have documents of varying length, longer documents may have higher scores for tf and idf. In order to compensate for this, the final score is normalized by the document length.

Given a document set S, for each term t in the vocabulary and a document $D \in S$, Okapi calculates the term frequency (tf) and inverse document frequency (idf) as follows:

tf weight: $w_{tf} = \frac{(k_1+1)tf}{k_1[(1-b)+b \times dl/avdl]+tf}$

idf weight: $w_{idf} = ln\frac{N-df+0.5}{df+0.5}$

Here tf is the frequency of occurrence of term t in document D; N is the total number of documents in the document set S; df is the number of documents in S that contain t; dl is the length of D (in terms); $avdl$ is the average length (in terms) of all the documents in S. b and k_1 are two predetermined constants. We use values of $b = 0.75$ and $k_1 = 1.2$ which are based on previous literature [90].

The relevance score between a document and a query is the inner product of the document vector and the query vector. Okapi defines the weight of a term in a query slightly differently from the weight in a document. However, to enable a symmetric comparison of two documents, $D_1 \in S$ and $D_2 \in S$, as discussed in [68] we use a single definition for the term weights for documents. We compute the similarity value between D_1 and D_2 as the the inner product of D_1's vector $\vec{V_1}$ and D_2's vector $\vec{V_2}$ as follows:

$$M_{sim}(\vec{V_i}, \vec{V_j}) = \sum_t w^1_{tf}(t) \times w^2_{tf}(t) \times w_{idf}(t)$$

where $w^1_{tf}(t)$ is the *tf*-weight of term t in vector $\vec{V}_1$, $w^2_{tf}(t)$ is the *tf*-weight of term t in vector $\vec{V}_2$, and $w_{idf}(t)$ is the *idf*-weight.

Let $Sim(p_i, p_j)$ be the similarity score of post p_i and post p_j. For similarity metric M_{sim},

$$Sim(p_i, p_j) = M_{sim}(\vec{V}_{p_i}, \vec{V}_{p_j})$$

where $\vec{V}_{p_i}$ is the term vector of post p_i, $\vec{V}_{p_j}$ is the term vector of post p_j.

3.2.2.2 User Validation of Similar Posts

Ideally, all similar posts in the ground truth for each query post would be identified by a human. Since we have several hundred query posts and tens of thousands of candidate future posts, (see details in Table 3.2), it would be very expensive to create the ground truth in this manner. We therefore used a compromise solution. We used a threshold of the Okapi similarity score to determine the ground truth posts; the threshold value is discussed in section 3.4. We then use human judgement to validate that the Okapi metric was indeed effective in differentiating the most similar posts from less similar posts, in the ground truth. We used Amazon's Mechanical Turk marketplace for this user validation.

We randomly selected 50 of the target query posts. For each query post, we selected 6 candidate posts from the 10-day test dataset (see section 3.4). 3 of the candidate posts had a high Okapi similarity score in the range of [120, 800] compared to the query posts; we label these as Group 1 of very similar posts. 3 of the candidate posts had a low Okapi score in the range of [40, 60], compared to the query posts; we label these as Group 2 of dissimilar posts. For each candidate post, we asked three users to evaluate the similarity between the candidate post and the query post. They were asked to rate the similarity using the following 4 values: *"very similar"*, *"similar"* , *"maybe similar"*, or *"not similar"*. To determine inter-annotator agreement, we assume that *"very similar"*, *"similar"* and *"may be similar"* represent one agreement, and *"not similar"* represents another agreement.

Table 3.1 reports on the ratings for each group. For the candidate posts in Group 1 (high similarity) 76% posts were ranked as *"very similar"* or *"similar"* by at least 2 users, and 93% of them were ranked as *"very similar"* or *"similar"* or *"may be similar"* by at least 2 users. For the candidate posts in Group 2 with low similarity scores, 8% of them were ranked as *"very similar"* or *"similar"* by at least 2 users, and 17% of them were ranked as *"very similar"* or *"similar"* or *"may be similar"* by at least 2 users. The inter-annotator agreement was 91% for posts with high similarity scores, and 83% for those with low similarity scores.

This validation confirms that the judgement of human users of the similarity between a candidate post and a query post is in agreement with the judgement based on the Okapi similarity scores.

3.2.3 Blog Channel Features

We consider several features. The first is the consistency of topics in a blog channel; we note that consistency is a factor in being an authoritative channel, since an authority on a topic will post more consistently on that topic than other topics.

Table 3.1: Percentage of the ratings of user evaluation

	Group1	Group2
ranked as *"very similar"* or *"similar"* by at least 2 users	76%	8%
ranked as *"very similar"* or *"similar"* or *"may be similar"* by at least 2 users	93%	17%
ranked as *"very similar"* or *"similar"* by all 3 users	57%	3%
ranked as *"very similar"* or *"similar"* or *"may be similar"* by all 3 users	82%	10%

We also consider named-entities, links between channels and links to external pages.

[77] has identified a blog distillation task as identifying blog channels that consistently and repeatedly post on the same topic(s) over time. If the content of a blog channel is very consistent, then we expect that it would be relatively easier to predict the topic of future posts.

We use the average of the pairwise similarity scores between different historical posts of a blog channel b to represent the consistency of blog channel b.

Formally, the consistency score $\psi(b)$ of blog channel b can be computed as follows:

$$\psi(b) = \frac{2}{m \cdot (m-1)} \sum_{p_i, p_j \in b, i \neq j} M_{sim}(\vec{V}(p_i), \vec{V}(p_j))$$

where p_i and p_j are historical posts of blog channel b, and m is the number of historical posts in blog channel b.

Figure 3.1 is a visualization of blog channel consistency. Each historical post is represented by a star symbol. The blog channel on the left is more consistent than the blog channel on the right in Figure 3.1. Visually, the historical posts are more closely clustered to each other for the more consistent blog channel on the left.

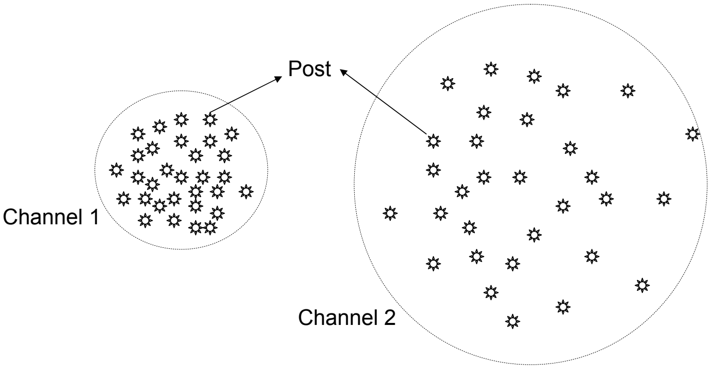


Figure 3.1: Blog channel consistency. The blog channel on the left is more consistent than the blog channel on the right.

Figure 3.2 reports on the distribution of the consistency scores for over 40,000 blog channels. There are 486 blog channels whose consistency scores are greater than 200 and 2997 blog channels whose scores are greater than 100.

3.2.4 Diffusion-Related Features

cRatio

The life cycle of a specific topic involves multiple stages of diffusion as visualized in Figure 3.3. The first stage is an *emerging* topic, e.g., at time T_1, with a

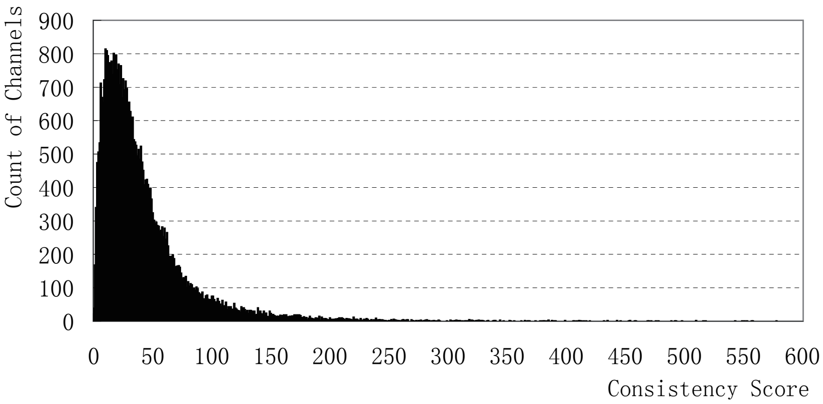


Figure 3.2: The distribution of blog channel consistency scores. The value on the Y axis is the count of the blog channels whose consistency score is (rounded off to) x, where x is the value on the X axis.

small number of people talking about it. Typically a topic reaches its peak with a relatively larger number of people talking about it, e.g., between T_2 and T_3. Approximately the same number of people may talk about it before or after the peak. Finally there is a *fading* with a few people (or no people) talking about it after T_4.

Gruhl et al. [40] studied the dynamics of information propagation in the blogosphere and proposed that topics are mostly composed of a union of chatter (ongoing discussions whose subtopic flow is largely determined by decisions of the authors) and spikes (short-term, high-intensity discussion of real-world events that are relevant to the topic). Usually for chatter, the shape will be more flat and the timespan is longer. For spikes, the shape may be more steep and the timespan may be shorter. Figure 3.4 visualizes the diffusion for several example topics from the Spinn3r dataset. 3.4(a) resembles spikes whereas 3.4(b) and 3.4(c) resemble chatter.

While there have been many mathematical models of diffusion [10]. We pro-

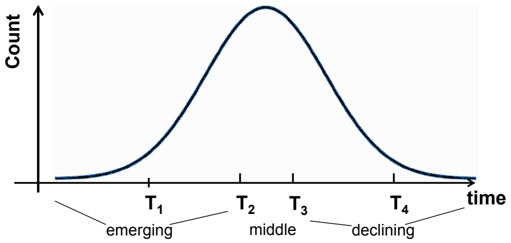


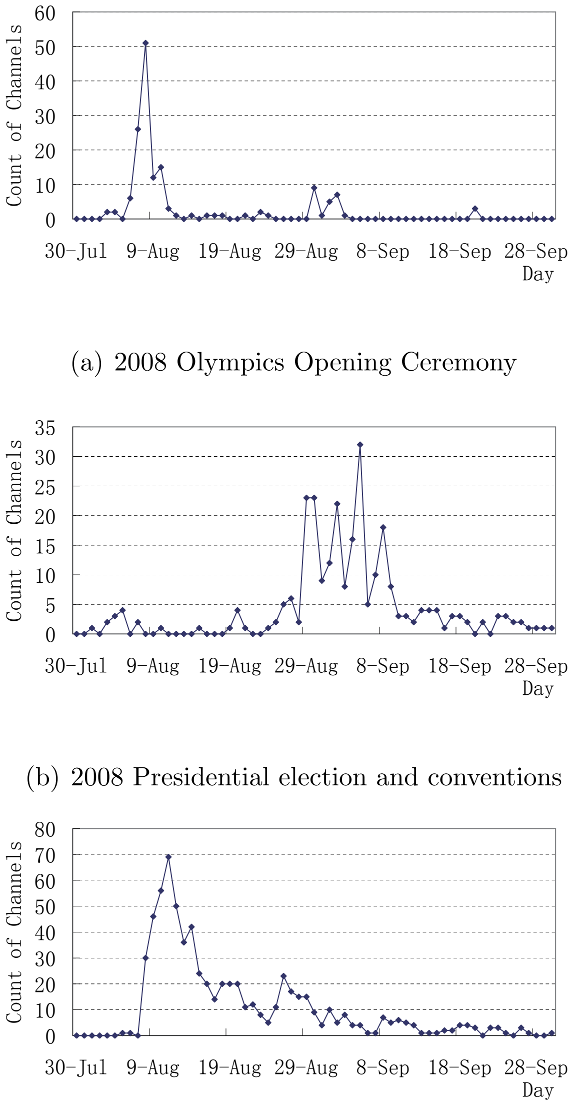
Figure 3.3: Diffusion stage of a topic. The value on the Y axis is the number of people talking about a topic at time T_i (X axis).

pose a simple metric *cRatio* to characterize the diffusion stage of a topic at time T and we will use this value of *cRatio* to prepare experiment datasets to reflect different stages of diffusion. Our experiments will show that the diffusion stage of a query post has a significant impact on prediction accuracy.

Consider a query post p in blog channel b at time T. Let N_{future} be the number of blog channels other than b with posts that are similar to p after time T. Let $N_{history}$ be the number of blog channels other than b with similar posts before T. Then, $cRatio = N_{history}/(N_{history} + N_{future})$ is used to represent the diffusion stage of the topic of post p at time T.

V/AC

Besides the diffusion stage, we find that the number of authors and posts on a topic generally affect prediction accuracy. To distinguish these topics, we define the concept of blog volume versus author count V/AC. For a query post p, suppose during the time period from T to $T + \Delta T$, there are N_{post} posts which are topically similar to p, and these posts come from N_{author} distinct blog channels. We define V/AC of query post p from T to $T + \Delta T$ as follows: $V/AC = N_{post}/N_{author}$.



(a) 2008 Olympics Opening Ceremony

(b) 2008 Presidential election and conventions

(c) 2008 South Ossetia war

Figure 3.4: Examples of diffusion of sample topics from our dataset. A query post represents a topic. The Y axis is the number of blog channels which contain posts similar to the query post on that day (X axis).

3.3 Prediction Methods

We develop several prediction methods. **PROF** and **VOTE** are inspired by information retrieval techniques and they are naive (they require no training). Maintaining a profile for a data stream has been addressed in [68, 86]. We build upon these ideas; **PROF** constructs a profile of historical posts (favoring recent posts) to represent a blog channel, and uses the profile to make a prediction. **VOTE** accumulates the vote of multiple historical and relevant posts to make a prediction. **RSVMP** uses a ranking SVM to exploit multiple features that were described in the previous section. It is a sophisticated and computationally expensive method since it requires training.

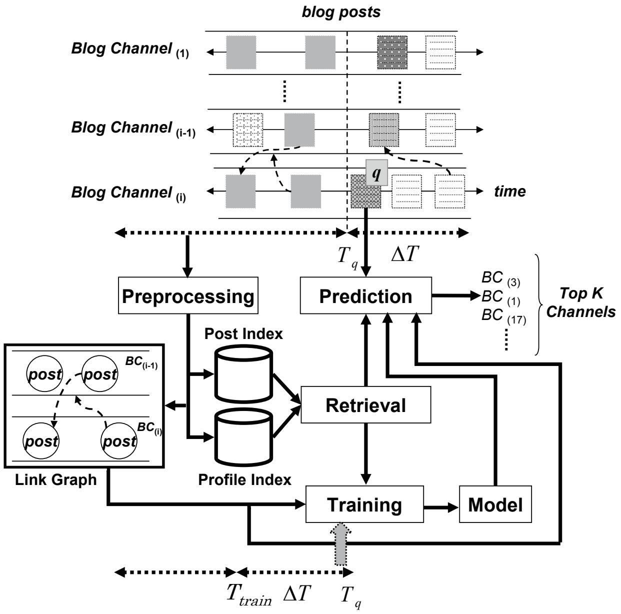


Figure 3.5: System architecture for prediction.

Figure 3.5 illustrates the basic components of our prediction methods. Given the query post q, all posts in a specific time window preceding T_q are preprocessed and the post index and the profile are built. The features such as the links between different blog channels and the links pointing to outside pages are extracted. The link graph is built. For *PROF* and *VOTE* methods, the indexes are used to retrieve a list of profiles or posts that are similar to q. Then these profiles or posts are processed to rank the blog channels. For the supervised method *RSVMP*, a set of training query posts are selected very close to T_{train}, where $T_{train} \leq T_q - \Delta T$. For each training query post, the most similar ground truth posts are retrieved, in the time interval $(T_{train}, T_{train} + \Delta T)$ to determine the *ground truth* blog channels. Further, for each training query post, and for each feature used to train the ranking SVM, the top K' ($K' \geq K$) matching blog channels are retrieved, based on the data before T_{train}; these blog channels are used for creating training pairs. Some of them are in the *ground truth* and some of them are not. To create the partial order of *training pairs*, for a training query post, a ground truth blog channel is ranked higher than a non ground truth blog channel. Then a model is trained. This model is used to make a prediction for a new coming query post. In a real time system, the training process should be repeated frequently to make the model reflect the recency.

3.3.1 Profile Based Prediction (PROF)

The profile of a blog channel represents the content of its posts and it should be updated as new posts appear. Maintaining profiles has been explored in several studies, e.g., in [68, 86]; the key issues include the number of terms to maintain and the frequency at which the profile is updated. A sliding window model is typically used to filter out stale information, but it sometimes misses relevant terms outside the window. Instead, we adopted a temporal decay model to update the profile. For simplicity, the decay model does not consider absolute time; instead, we treat the time interval between updates as a time unit.

Suppose $\{p_1, p_2, ..., p_n\}$ is a sequence of posts in blog channel b and each post p_i is represented as a weighted term vector $\vec{V}_{p_i}$.

The blog channel profile vector $\vec{V}_b^1$ is initially set to $\vec{V}_{p_1}$ upon arrival of post p_1. As each new post p_i arrives at time unit i, the blog channel profile vector $\vec{V}_b^{i-1}$ is updated to $\vec{V}_b^i$ as follows:

$$\vec{V}_b^i = \theta \cdot \vec{V}_b^{i-1} + (1 - \theta) \cdot \vec{V}_{p_i} \tag{3.1}$$

θ is a temporal decay factor, $0 < \theta < 1$; we choose an appropriate value for θ based on tuning from experiment datasets. We treat the profile of a blog channel as a document [2].

After the profiles are built, they can be indexed. The profile based prediction algorithm is to retrieve the top blog channels ranked by the their similarity scores to the target query post. The similarity of the profile of channel b to query post q is $Sim(q, b)$ and it is computed as follows:

$$Sim(q, b) = M_{sim}(\vec{V}_q, \vec{V}_b^n)$$

PROF uses $Sim(q, b)$ to retrieve the Top K blog channels.

[2] For the convenience of indexing the profile of a blog channel, we treat the profile of a blog channel as a document. As the weight of each term in a blog channel profile is more likely to be a decimal value, for implementation simplicity, we wanted to transfer it to be an integer value and at the same time keep enough precision. In experiments, we multiplied the weight of each term by 10 and then rounded it up to an integer.

The detail of **PROF** is described in Algorithm 1.

```
Algorithm 1 Profile Based Prediction (PROF)
Input: Profile Index, query post $q$, $K$
Output: Top K blog channels
1. Query the profile index. For each blog channel $b$ whose profile contains any common terms with query post $q$, compute the similarity score $Sim(q, b)$.
2. Select top $K$ blog channels with the descending order of their similarity scores with query post $q$.
```

3.3.2 Voting Based Prediction (VOTE)

VOTE chooses the top K channels using the aggregate similarity score of all historical posts in a channel b. For a given query post q, the aggregate similarity score for channel b is the sum of all similarity scores of posts $p_i \in b$ as follows:

$$score(q, b) = \sum_{p_i \in b} M_{sim}(\vec{V}_q, \vec{V}_{p_i})$$

VOTE restricts the score to consider only the Y ($> K$) most similar posts and returns the top K channels.

The method is described in Algorithm 2.

3.3.3 Ranking SVM Based Prediction (RSVMP)

A ranking SVM was trained to predict the Top K author blog channels. We briefly review a ranking SVM and then discuss feature selection and the ground truth training data for this task.

Algorithm 2 Voting Based Prediction (**VOTE**)

```
Input: Post Index, query post $q$, $K$, $Y$ ($Y > K$)
Output: Top K blog channels
1. Query the post index to obtain list $L_Y$ of the top $Y$ posts with the highest similarity scores to the query post $q$.
2. For each post $p_j$ in step 1 that occurs in blog channel $b_i$, $score(q, b_i) = \sum_{p_j \in b_i, p_j \in L_Y} M_{sim}(\vec{V}_q, \vec{V}_{p_i})$.
3. Sort the blog channels by $score(q, b_i)$. Return at most $K$ blog channels in descending order.
```

Ranking SVM:

We represent the match of a blog channel to a query post as a vector $\vec{x}$. Each element in the vector is a numerical value indicating some correlation between the blog channel and the query post. There are different types of correlation between a blog channel and a query post and hence there are multiple elements in a vector $\vec{x}$. Any pair of vectors $(\vec{x}_i, \vec{x}_j) \in$ a ranking R if $\vec{x}_i$ ranks higher than $\vec{x}_j$ in R. Suppose that there is some optimal ranking R^* representing the ground truth. The goal (of the ranking SVM) is to find a ranking function f that approximates the optimal ranking R^*. A ranking function f is evaluated by comparing its ranking R^f with R^*. Kendall's τ is the most frequently used metric to compare two rankings [51]. We denote the Kendall's τ between some R^f and R^* as $\tau(R^f, R^*)$.

In practice, the optimal ranking R^* is not available. The ranking SVM is provided with training data corresponding to one or more partial rankings (partial orders) $R' \in R^*$. It can then learn a ranking function f from these partial orders.

To be specific, we apply a generic mapping from the feature vector $\vec{x}$ in the original feature space to a new feature vector $\phi(\vec{x})$ in a virtual feature space. When $\phi(\vec{x}) = \vec{x}$, the SVM kernel is linear. Assume f is a ranking function as follows:

$$\forall(\vec{x}_i, \vec{x}_j) \in R' : f(\vec{x}_i) > f(\vec{x}_j) \Longleftrightarrow \vec{w} \cdot \phi(\vec{x}_i) > \vec{w} \cdot \phi(\vec{x}_j) \tag{3.2}$$

The goal is to learn an f which is concordant with the given partial orders $R' \in R^*$ and which can also generalize well beyond R'. One approach is to determine $\vec{w}$ that satisfies equation (3.2) for the maximum number of pairs of elements $(\vec{x}_i, \vec{x}_j) \in R^*$ while simultaneously maximizing $\tau(R^f, R^*)$. This problem is known to be NP-hard [24]. A ranking SVM will obtain an approximate solution by solving the following optimization problem [43]:

$$\mathit{minimize:} \qquad \frac{1}{2}|\vec{w}|^2 + C\sum \xi_{i,j}. \tag{3.3}$$

subject to:

$$\forall(\vec{x_i}, \vec{x_j}) : \xi_{i,j} \geq 0 \tag{3.4}$$

$$\forall(\vec{x_i}, \vec{x_j}) \in R' : \vec{w}(\phi(\vec{x_i}) - \phi(\vec{x_j})) > 1 - \xi_{i,j} \tag{3.5}$$

$\xi_{i,j}$ are non-negative slack variables to allow some training error. C is a parameter that controls the trading-off between the margin size and training error, The solution weight vector can be written in the form of training pairs as [43]:

$$w^* = \sum \alpha^*_{i,j} t_{i,j} (\phi(x_i) - \phi(x_j))$$

where $\alpha^*_{i,j}$ can be computed by kernel function of training pairs [43]. $t_{i,j} = +1$ if $(\vec{x_i}, \vec{x_j}) \in R'$; $t_{i,j} = -1$ if $(\vec{x_j}, \vec{x_i}) \in R'$. For the case of a linear kernel, w^* can be computed explicitly, which makes the ranking function just a linear combination of feature weights.

The ranking SVM will then use w^* for prediction, to rank the set of candidate blog channels, for some incoming query post.

Training Pairs:

Assume we only consider the ground truth in the future time span ΔT. To predict top K author blog channels for a query post q at current time T_c, the training pairs are chosen as follows:

Select N_{train} training query posts with posting time near T_{train}, where $T_{train} \leq T_c - \Delta T$. Get the ground truth of each training query post in the time range $(T_{train}, T_{train} + \Delta T)$. For each training query post, retrieve top K' $(K' \geq K)$ blog channels by each SVM feature from the data before T_{train}. All of these blog channels are considered to be the candidate blog channels of that training query post. Collect the SVM features of the candidate blog channels of each training query post from the data with posting time before T_{train}. For a training query post, a candidate blog channel which is in the ground truth is set to be ranked higher than a candidate blog channel which is not in the ground truth. Each pair of them is composed to be a training pair.

Feature Selection and Training Data for RSVMP:

Consider a candidate blog channel b and a query post q.

- *FT-CHANNEL*: The similarity score between q and the profile of blog channel b.
- *FT-POST*: The similarity score between q and the post in b which is most similar to q.
- *FT-NE*: The similarity score between a document composed of all named-entities extracted from q and the profile of blog channel b.
- *FT-PROFILE*: The similarity score between the profile of the author blog channel for q and the profile of blog channel b.
- *FT-CONSISTENCY*: The consistency score for blog channel b.
- *FT-OFFLINKS*: The (weighted) count of offsite links that are common to the author blog channel for q and blog channel b.
- *FT-INSIDELINKS*: The (weighted) count of channel to channel links between the author blog channel for q and blog channel b.

We compute tf and *idf* values for the feature *FT-OFFLINKS* as follows: For a common offsite link URL_a, suppose URL_a appears N_1 times in blog channel b and N_2 times in the author blog channel for q. Suppose that there n blog channels that also contain URL_a. Then the weighted contribution to the score *FT-OFFLINKS* by $URL_a = \frac{N_1 \cdot N_2}{n}$. The total score for the feature *FT-OFFLINKS* is the summation of all scores contributed by the common offsite links between the two channels. To compute the value of the feature *FT-INSIDELINKS* we divide the count of links between the two blog channels by the total count of links between the blog channel b and all other blog channels in the training dataset.

3.4 Experimental Evaluation

3.4.1 Evaluation Datasets and Metrics

3.4.1.1 Dataset

The dataset provided by Spinn3r.com is a set of 44 million blog posts crawled between August 1st and October 1st, 2008. The post includes the document content as well as metadata such as the blog channel homepage, timestamp, title, category keywords, etc. The data is formatted in XML. The total size of the dataset is 142 GB uncompressed, (27 GB compressed). We extracted and processed a subset of posts in English. The total number of English posts is 13.87M, and the total number of English blog channels is 894K. Half of the these blog channels contain no more than 2 posts and some blog channels contain a large number of posts; the maximum number of posts for one English blog channel is 152K.

We focus on blog channels with human authors rather than machine generated posts. We note on inspection that blog channels with a high frequency of posts in the interval were often machine generated or were other kinds of information channels rather than real blog channels. We created a dataset with at least one post per two days as follows: We selected the posts that were published between July 30 and October 1 2008, the interval of interest. We then filtered out the blog channels that have less than 30 posts or more than 120 posts in the interval of interest. The statistics of the dataset that was used for the evaluation is in Table 3.2.

Table 3.2: Statistics of the blog channel experiment data set

Time range	07/30/08–10/1/08
Number of blog posts	2,185,810
Number of blog channels	42,005
Avg number of posts per blog channel	52.04

3.4.1.2 Query Posts and Ground Truth

We created 2 sets of query posts, Q_1 and Q_2; these query posts are obtained from the beginning of an interval starting on September 1. We created three test datasets to obtain ground truth posts for Q_1 and Q_2. One test dataset included 2 days of posts from September 1 to September 2, another included 10 days of posts from September 1 to September 10, and a third included a 30 day dataset from September 1 to 30. A ground truth blog channel is one that includes at least one future post (in some test dataset) that is similar to the query post (in Q_1 or Q_2). We used an Okapi similarity score of 130 as the threshold to identify ground truth posts.

For example, Q_1 contains 861 query posts. Each focal post matched an average of 22 ground truth blog channels in the 2-day test dataset and 47 in the 10-day test dataset. The query posts in Q_2 had similar numbers of ground truth blog channels. Figure 3.6 reports on the distribution of the number of ground truth blog channels for the 861 focal query posts in Q_1. We note that a small number of the focal query posts have a large number of ground truth blog channels while a large number have a

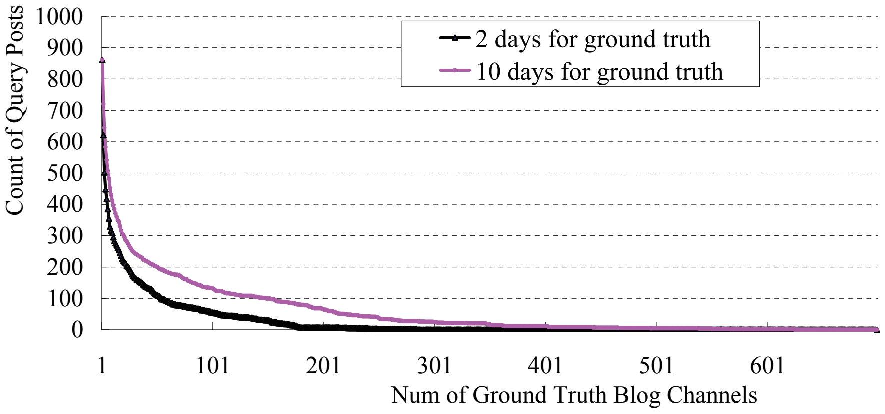


Figure 3.6: The distribution of the number of ground truth blog channels. The value on the Y axis is the count of the focal query posts whose ground truth contains at least x blog channels, where x is the value on the X axis. One curve is for the 2-day test dataset and another curve is for the 10-day test dataset.

small number of ground truth blog channels. This is consistent with the well known power law characteristic.

3.4.1.3 Subset of Query Posts and Ground Truth

Recall that consistent blog channels (Section 3.2.3), and the diffusion stage of topics indicated by *cRatio* and V/AC (blog volume versus author count) (Section 3.2.4), may all impact prediction accuracy. To test this, we created different subsets of query posts based on different values of *cRatio* and V/AC (blog volume versus author count). We also created a subset of consistent ground truth blog channels. The subsets are as follows:

- Subset of query posts in Q_1 and Q_2 with different ranges of *cRatio* values.

The query posts were separated into 6 groups, with *cRatio* in the following ranges: $[0, 0.2]$, $(0.2, 0.4]$, $(0.4, 0.5]$, $(0.5, 0.6]$, $(0.6, 0.8]$, $(0.8, 1.0]$.

- Subset of consistent blog channels with consistency scores in the range of $[60, +\infty]$. There were 8300 such channels. We used the query posts from Q_2 and filtered the 10 day test dataset to only include the similar posts from these consistent blog channels, i.e., only the consistent blog channels containing similar posts were considered as the ground truth of the focal query posts in Q_2.
- Subset of query posts having V/AC in the range $[1.5, +\infty)$. We created a subset of query posts from Q_1 and a subset of query posts from Q_2 with the value of $V/AC \geq 1.5$. The ground truth for this high V/AC was calculated in the 10-day test dataset.

3.4.1.4 Training Data

The training data was obtained from July 30 to August 31. All historical posts in this period, for each blog channel, were used by both **VOTE** and **PROF** in a straightforward manner. For **RSVMP**, we selected the training query posts at the start of an interval on August 22. We created 2 ground truth datasets. The first used posts that occurred within 2 days after August 22, and the second used posts that occurred within 10 days. The features of the blog channels that were used to produce the training pairs for each training query post were collected in the interval from July 30 to August 21. We reiterate that there was *no overlap* between the training data and testing data.

3.4.1.5 Metrics and Parameters

Precision, recall, and the F measure are set-based measures computed on unordered sets of documents. Mean Average Precision (MAP) is widely used for evaluating ranking methods. It provides a single-figure measure of quality across recall levels. MAP has good discrimination and stability [69]. We use the Wilcoxon signed-rank test [106] to determine statistical significance. We also report on $P@1$, the precision of the Top 1 prediction, for the authoring task.

We set the temporal decay factor $\theta = 0.8$ for building blog channel profiles by tuning the training data. The K value of top K was set to 1000.

3.4.2 Experimental Results

Our first experiment shows the performance for the *authoring (AT)* task and the next reports on the *FAP* task. For both *AT* and *FAP*, we consider a subset of consistent blog channels as well. We also report on factors that improve prediction accuracy, and our confidence in the prediction. This includes the diffusion stage, measured by different values of *cRatio*, and the V/AC ratio, for the *FAP* task.

3.4.2.1 Baseline Results for the AT Task

Figure 3.7(a) reports on the MAP for the 3 methods on two test datasets. The label *"Entire Dataset"* corresponds to focal query posts from Q_1; the ground truth blog channels is from the 10-day test dataset. The label *"High Consistency"* corresponds to Q_2 and the 10-day test dataset. These blog channels were filtered to only include consistent blog channels.

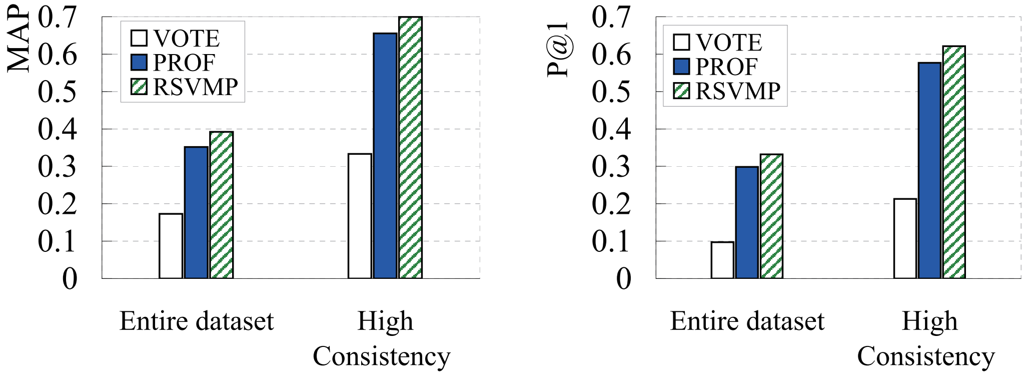
For the "Entire Dataset", *RSVMP* has a *MAP* value of 0.39, and *PROF* has an MAP of 0.35. Given that there is only one ground truth author for any post, these MAP values are surprisingly good, reflecting an accurate prediction. For the "High Consistency" channels, all methods show increased accuracy as expected. *MAP* is as high as 0.70 for *RSVMP*. This suggests that our methods perform with good accuracy on the *AT* task. We note that only 4 of the 7 correlation features were useful for this prediction; they are *FT-CHANNEL*, *FT-POST*, *FT-NE* and *FT-CONSISTENCY*.

Since there is only one author per post, Figure 3.7(b) reports on P@1, for all 3 prediction methods for the 2 datasets. As expected, these values are not as high as MAP. Nevertheless, they reflect a reasonable quality of prediction.

The Wilcoxon signed-rank test on *MAP* shows that *RSVMP* significantly outperforms *PROF* and *PROF* significantly outperforms *VOTE*, all with p far smaller than 0.01. We further note that while *RSVMP* can benefit from the training data, the improved accuracy of supervised learning over a naive *PROF* is limited.

3.4.2.2 Baseline Results for the *FAP* Task

Figure 3.8 reports on the MAP for the *FAP* Task for the 3 methods. The test datasets labeled as *"Entire Dataset"* and *"High Consistency"* are the same as



(a) MAP

(b) P@1

Figure 3.7: The performance for the *AT* task.

was used for the *AT* task. Unlike the *AT* task, where all 3 methods had reasonable prediction accuracy and where *PROF* and *RSVMP* showed very good performance, the *FAP* task is much more challenging. For the "Entire Dataset", *RSVMP* has the best *MAP* value of 0.23 while *PROF* has a value of 0.20. For "High Consistency", the *MAP* increases to a value of 0.42 for *RSVMP*. We note that while these MAP values may appear to be low, they are comparable to the MAP values reported for the TREC blog distillation task [77]; there the reported MAP values are also in the range of 0.10–0.30. The Wilcoxon signed-rank test shows that *RSVMP* significantly outperforms *PROF* and *PROF* significantly outperforms *VOTE*, all with p far smaller than 0.01.

Table 3.3 reports on the MAP of all the methods, for focal query posts in Q_1, w.r.t. the test datasets of different time spans, for the FAP task. Prediction accuracy for the 10-day test dataset is higher. This is probably because of the greater number of ground truth blog channels.

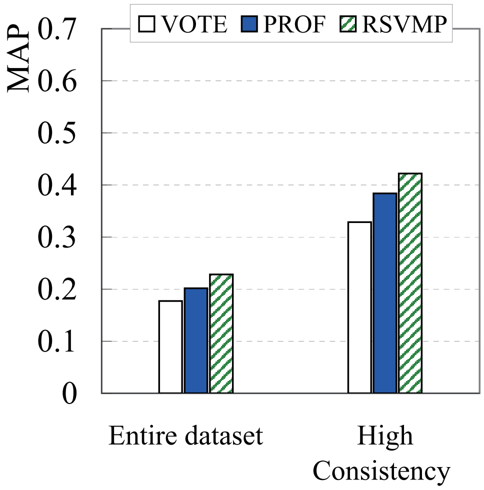


Figure 3.8: The performance for the *FAP* task.

Table 3.3: MAP for the *FAP* task w.r.t. different test datasets.

	VOTE	PROF	RSVMP
2-day test dataset	0.1383	0.1474	0.1672
10-day test dataset	0.1773	0.2018	0.2281

3.4.2.3 Impact of Diffusion Stage (cRatio Values)

Recall that $cRatio = N_{history}/(N_{history} + N_{future})$. N_{future} is the number of blog channels other than b with similar posts after T in the 30 day test dataset. $N_{history}$ is the number of blog channels other than b with similar posts before T in the 30 day training dataset.

Table 3.4 reports on the MAP values for the *FAP* task, for the 3 methods, for the focal test query posts from Q_1. The ground truth is from the 10-day test dataset. The results are grouped by the *cRatio* values for the query posts. Table 3.5 reports on the MAP for the same methods for the focal query posts from Q_2.

Table 3.4: The impact of *cRatio* on the "Entire Dataset" for the *FAP* task.

cRatio	0-0.2	0.2-0.4	0.4-0.5	0.5-0.6	0.6-0.8	0.8-1.0
VOTE	0.090	0.167	0.234	0.222	0.137	0.062
PROF	0.144	0.193	0.257	0.244	0.151	0.056
RSVMP	0.188	0.225	0.288	0.262	0.170	0.070

Table 3.5: The impact of *cRatio* on the "High Consistency" test dataset for the *FAP* task .

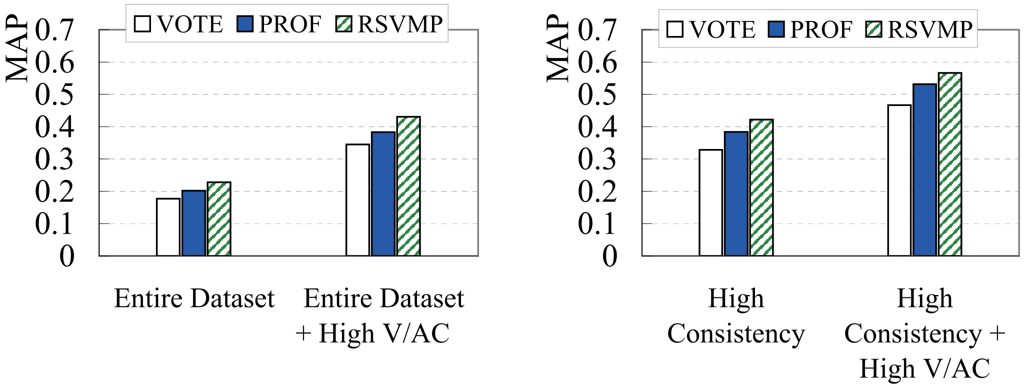
cRatio	0-0.2	0.2-0.4	0.4-0.5	0.5-0.6	0.6-0.8	0.8-1.0
VOTE	0.091	0.233	0.498	0.437	0.228	0.110
PROF	0.172	0.285	0.578	0.487	0.262	0.214
RSVMP	0.205	0.309	0.605	0.525	0.314	0.235

The ground truth is from the consistent blog channels in the 10-day test dataset.

PROF outperforms *VOTE* and *RSVMP* dominates both. The value of MAP is highest for all the methods when *cRatio* is in the range 0.4–0.5 and 0.5–0.6, i.e., the middle stage of diffusion. When *cRatio* is in the range 0.4–0.5, and for consistent blog channels, *RSVMP* has an MAP value that is as high as 0.61. *RSVMP* also does surprisingly well for emerging topics.

3.4.2.4 Impact of Blog Volume Versus Author Count

Figure 3.9(a) reports on the MAP for the 3 methods for the *FAP* task. The left part reports on the *"Entire Dataset"* and the right reports on *"Entire Dataset"*



(a) Entire Dataset

(b) High Consistency

Figure 3.9: The impact of V/AC for the *FAP* task.

with high values for V/AC. Figure 3.9(b) reports on the MAP for consistent blog channels. The left part reports on the *"High Consistency"* and the right reports on *"High Consistency"* with high values for V/AC.

With increased values of V/AC, prediction accuracy improves, across all methods and across all datasets. This is consistent since high V/AC reflects repeated posts by some authors, thus making the *FAP* prediction task somewhat easier. Further, *RSVMP* has an MAP value of 0.57 in Figure 3.9(b) for the *"High Consistency"* test dataset with high values for V/AC. This reflects that we can predict repeated posts on the same topic by consistent authors, with high prediction accuracy, or high confidence in the prediction.

3.4.2.5 Difficult and Diverse Predictions

We note from the previous discussion that *RSVMP* is able to exploit multiple features and provide a more accurate prediction even in a difficult prediction scenario corresponding to an emerging topic. A further analysis of the properties of the predicted blog channels of *PROF* and *RSVMP* illustrate that the predictions of *RSVMP* may have high utility. For example, we compared the consistency scores of the Top 10 predictions for the two methods over the entire dataset. The average score is 108.6 for *RSVMP* and 112.7 for *PROF*. We also compared the average profile similarity scores between the predicted blog channels and the focal query post over the entire dataset. The average score is 399 for *RSVMP* and 428 for *PROF*. Thus, *RSVMP* was able to successfully identify the *less consistent* authors who have not posted on the focal topic in the past but who will post on the topic in the future. Similarly, *RSVMP* was able to successfully identify authors whose profile was not similar to the focal query post, but who nevertheless authored a post that was similar to the focal post. To summarize, these less consistent authors or authors with dissimilar profiles who nevertheless will post on the focal topic in the future may have more utility for the task of monitoring.

Chapter 4

Future Link Prediction in the Blogosphere

4.1 Introduction

The amount of new content and links being generated within social media, including blogs, micro-blogs and user-generated content, is increasing dramatically. Users are becoming content producers or bloggers (authors), either by themselves or as a team; they then populate their blogs with a stream of posts (blog entries). When creating their posts, bloggers use hyperlinks to refer to a variety of pages and websites including the posts of other bloggers and more often, other blog channels. This collection of blogs, i.e., the blogosphere, can be viewed as a dynamically changing representation of the evolution of content streams, with an overlay of links. These links, in turn, can be viewed as a proxy to indicate the direction of information flow and influence in the blogosphere [40, 53, 93].

On the user side, a consumer of blogs might see a stream of interesting posts, and wonder if there are other blog channels that will link to this focal blog channel in the future. Such a link will indicate that the other blog channel is interested in a similar topic. For a creator of a post on a focal channel, it is important to inspire a conversation around a particular topic, and so they want to know who will link to them. In both of these cases, the concern is *who will link to a focal blog channel in the near future?* This question is important because bloggers and fans need to prioritize the blogs that they may visit. A link from one blog channel to another often indicates that the blogger is following the topics that are being discussed in the other blog channel. Similarly, a link could indicate that the blogger is a fan of the author of the focal blog channel, is influenced by this author, or is creating relevant content.

The timeframe of prediction is an additional critical factor given the stream of content in the blogosphere. Recently published posts typically attract more readers than older posts. However, recent posts may not have had sufficient time exposure to attract many links. Hence, accurate future link prediction is an important element when making recommendations for recent posts.

Future link prediction can also be a key element of a successful word-of-mouth (WOM) marketing strategy. WOM marketing refers to using a consumer's existing social network to encourage the passing of information from peer to peer and can include blogs and other social media. The effect of peer information on a consumer who is making a purchase decision is significant [39] because the receivers of WOM referrals tend to believe that their peers are more honest. More influential peers usually have a disproportionate effect on their friends when it comes to product adoption. Thus, for WOM marketing to be a success, a marketer has to identify influential blog channels. Two important factors which affect the influence of a blog channel are the content posted on the blog channel and the links pointing to the blog channel or its posts. The presence of links can directly increase the status of the blog channel since inlinks are often considered to be a "vote of confidence"; this is most significant when the inlink comes from an influential blogger.

Since understanding the future state of the link structure of the blogosphere can help bloggers, consumers of blogs and marketing managers, in this chapter, we address the problem of *future link prediction (FLP)*. Informally, FLP is as follows: Given a focal blog channel, predict the Top K blog channels that will contain at least one future post that will link to the focal blog channel or some post in it. A solution to FLP will also answer the following question: Can we identify those channels that will get at least one future link?

FLP within the blogosphere is novel and challenging. The graph that represents the blogosphere continuously evolves in many ways. If each node represents a stream of events or blog posts, then the nodes change with each event, i.e., each new post. Further, the edges in the graph also evolve. While a solution to FLP may use historical links, we note that these are often sparse and therefore difficult to utilize. FLP is also challenging since we want to predict future links in a finite time interval following the event of each new blog post. FLP is more difficult than static link prediction where the goal is to indicate whether or not a future link will exist. Thus, FLP in social media is characterized by the following features: (1) a sparse network of historical links; (2) a graph where both node features and links evolve; (3) a temporal profile for nodes that evolves with each event (new blog post); (4) the need to make a prediction in a finite time window shortly following the event (new blog post).

There are two approaches to link prediction that have been successfully applied in other contexts [59, 64]. One approach is content-based, i.e., we compare the content of the focal post to the content of all other blog channels and choose the one with the closest content match. The second approach is network-based and utilizes the structural properties of the focal blog channel and the other blog channels to infer missing structure. At times these two viewpoints seem at odds; network science has often suggested the dominance of the structure of a network. Our view is that these approaches are not mutually exclusive but rather they are two ends of a spectrum. We believe that a hybrid of structural- and content-based properties is needed to make accurate predictions in the blogosphere.

To examine this hypothesis, we apply several topological metrics that use historical links for prediction, including *Jaccard*, *CommonNeighbors*, and *Bonacich*. To efficiently calculate *Bonacich*, we present a method *Bonacich-A* based on an approximate *Bonacich* score. Moreover, we incorporate an additional network metric; *CommonExternal* is a method based on external links. Besides link features, we also explore content features. We examine a content based method *CBP*; it uses a temporal profile to represent the interests and content of a blog channel based on historical posts. In addition, we propose a simple unsupervised learning based hybrid solution *HYBRID* that combines the features of *Bonacich-A* (the best solution among the link-based prediction methods) and *CBP*. Finally, we present a supervised learning method *RSVMP*, which is a ranking support vector machine for FLP.

We report on the results of an evaluation on a blog dataset from Spinn3r. Among the network-based methods, *Bonacich-A* has the best prediction accuracy. *CommonExternal* and *CommonNeighbors* have similar accuracy and this demonstrates the importance of external links as a feature for prediction. While the content-based method *CBP* has the lowest prediction accuracy, it can significantly improve both hybrid solutions, *HYBRID* and *RSVMP*. The supervised learning method *RSVMP* can use all the features and has the highest prediction accuracy. It does surprisingly well for datasets when there are no historical links; the benefit from the *CPB* feature is particularly useful. *RSVMP* is most accurate for FLP over a short time interval (10 days) of prediction. These results are presented in [110].

4.2 Problem Definition

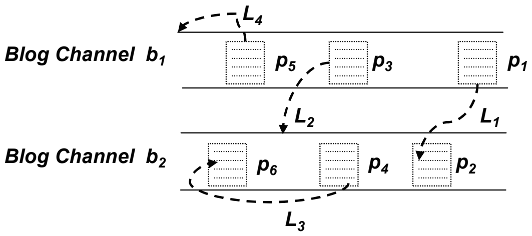


Figure 4.1: Links in the blogosphere

We define a blog channel as an event stream of posts (i.e., blog entries) originating from a single source (a blogger, organization, etc.). Figure 4.1 depicts two blog channels with three posts and links between them.

A link in a post that points to another post is a "post-to-post" link. Link L_1 in the figure is an example and points from post p_1 to post p_2. A link in a post that points to a blog channel is a "post-to-blog-channel" link; L_2 from post p_3 to blog channel b_2 is an example. We abstract both types of links as "blog-channel-to-blog-channel" links. If a link points to the focal blog channel in which it appears, or to another post of the focal channel, then it is a “self-referential” link; L_3 and L_4 are examples.

Definition 2 ***Future Link Prediction (FLP) Problem:*** *Given a focal blog channel b at a specific time point T and a time interval ΔT, identify the blog channels $B^b_{T,\Delta T}$ that will contain one (or more) future post(s) in the interval $(T, T + \Delta T]$ having at least one link pointing to blog channel b or any post of blog channel b. We consider a simplified problem to identify up to K blog channels in $B^b_{T,\Delta T}$.*

Assume that all historical data in the period that precedes T is available. We can then identify other blog channels with historical links to blog channel b; they are the followers of b. Though the exact state of the network evolves over time, we also know that social connections are persistent. Thus, future links between different blog channels can be inferred from historical links. In other words, historical followers may continue to place links in the future. The follower relationship also tends to be transitive. If b' is a follower of b, then a follower of b' could become a follower of b. In addition, each blogger has particular interests. Other blog channels that share similar interests with blog channel b may be more likely to place links to b. Shared interests may be determined using historical posts. We will use these two aspects: (1) the consistency of the social network, and (2) shared interests to construct solutions to the FLP problem.

4.3 Prediction Methods

4.3.1 Link Based Prediction Methods

If a post in blog channel b_j refers to a blog channel b_i or to posts in b_i, then blog channel b_j is influenced by or is following b_i and is more likely to refer to b_i in the future. We further expect that this property of b_j following b_i should be transitive. We consider blog channels that contain direct links to the focal blog channel and indirect links (or paths).

We create a *blog channel link graph* using historical links and posts. Nodes represent blog channels and edges represent links between blog channels. The *blog channel link graph* is a directed graph. An edge is weighted by the count of the number of inlinks to the focal blog channel.

We apply the methods surveyed in Liben-Nowell and Kleinberg [64] to create structural metrics to the blogosphere:

- **Jaccard:** This coefficient is commonly used in information retrieval [88]; it measures the number of features that i and j have in common, compared to the number of features of either i or j. The "features" here are the neighbors in the graph. For a node i, let $\Gamma(i)$ be its set of neighbors. The score of nodes $score(i, j) = |\Gamma(i) \cap \Gamma(j)|/|\Gamma(i) \cup \Gamma(j)|$.

- **CommonNeighbors:** This method will directly use the count of common neighbors [74]; $score(i, j) = |\Gamma(i) \cap \Gamma(j)|$.

- **Bonacich-A:** Katz [52] measures the status of an node by the total number of paths linking it to other nodes in the graph; an exponential discount is used as the path length increases. Bonacich [15] generalized Katz's metric and proposed *Bonacich centrality.* It too reflects the total number of paths originating from a node and uses an attenuation factor α to discount indirect links and β to discount direct links. We briefly review the *Bonacich* metric and propose a prediction method, **Bonacich-A**, based on an approximate *Bonacich* score.

Let A be the adjacency matrix for the *blog channel link graph.* Recall that each node in the link graph is a blog channel. Suppose T is the time when the future link prediction is to be made. Thus, we consider all historical links prior to T. We set the value of the element $A_{i,j}$ in the link graph adjacency matrix A to be the number of direct links pointing from node j to node i that exist in history before time T. Then, the value of $(A^n)_{i,j}$ is equal to the number of paths of length n from node j to node i in history.

The Bonacich metric $(C(\alpha, \beta))_{i,j}$ reflects the influence on node j from node i in history. Bonacich centrality is computed as follows:

$$\begin{aligned} C(\alpha, \beta) &= (\beta A + \beta\alpha A \cdot A + ... + \beta\alpha^n A^{(n+1)} ...) \\ &= \beta A(1 - \alpha A)^{(-1)} \end{aligned}$$

This equation holds while $\alpha < 1/\mu$, where μ is the largest characteristic root of A [34]. For $\alpha = \beta$, this measure reduces to the Katz score.

Bonacich-A Method is described as Algorithm 3.

To efficiently calculate all of $(C(\alpha, \beta))_{i,j}$ for all j, we only need to consider the

Algorithm 3 Bonacich-A Method

```
Input: Blog channel link graph, Focal blog channel $b$, Adjacency matrix $A$, $K$, $\alpha$, $\beta$, $D$
Output: (Up to) Top $K$ blog channels
1. Let $i$ denote the node of the blog channel $b$. Compute the value of $(C(\alpha, \beta))_{i,j}$ for all other blog channels $j \neq i$ up to path of length $D$.
2. Rank all blog channels $j$ based on the value $(C(\alpha, \beta))_{i,j}$.
3. Select up to $K$ blog channels with nonzero values of $(C(\alpha, \beta))_{i,j}$.
```

subset of the link graph whose nodes have paths to i. We can then use a smaller adjacency matrix to calculate the Bonacich centrality. All of the nodes which have paths to i can be obtained by a breadth first search starting from i and traversing through the reverse direction of the edges of the *blog channel link graph*. The result will be identical to the solution obtained using the adjacency matrix of the whole *blog channel link graph*. In step 3, we exclude those j with $(C(\alpha, \beta))_{i,j}$ equal to 0. When $(C(\alpha, \beta))_{i,j}$ is equal to 0, it means j does not have any paths to i. We use parameter D to approximate the computation. In general, lower values of D will have a significant impact on the values of $(C(\alpha, \beta))_{i,j}$ since it is an approximate computation. For the sparse blog-channel-to-blog-channel link graph, D did not have much impact.

The link prediction methods **Jaccard**, **CommonNeighbors** and **Bonacich-A** are all based on links within the dataset or between different blog channels. Many links point to pages outside the blog dataset. We propose the following method based on external links and inspired by the TF/IDF metric popular in information retrieval.

- **CommonExternal:** For each external link URL_a that is common to blog channels i and j, suppose the link appears N_i (N_j) times in the corresponding blog channel i (j). Suppose that there are B_a blog channels that contain URL_a. Then, the weighted contribution to $score(i,j)$ by $URL_a = \frac{N_1 \cdot N_2}{B_a}$. The final score $score(i,j)$ is the summation of all scores contributed by all common external links between the two channels.

4.3.2 Content Based Prediction Method (CBP)

Historical links are very likely to be the best predictor of future links. However, such links are often sparse. Thus, it is important to consider additional features. The historical posts of each blog channel can be used to construct a profile to represent the content of that blog channel. Our content based prediction (CBP) method is based on the intuition that a blog channel is more likely to link to other blog channels which are similar in content. CBP uses the blog channel profile and a similarity metric to make a prediction.

4.3.2.1 Blog Channel Profile

The profile of a blog channel represents the content of its posts and it should be updated as new posts appear. Maintaining profiles has been explored in several studies, e.g., [86]; the key issues include the number of terms to maintain and the frequency by which the profile is updated. A sliding window model is typically used to filter out stale information, but it sometimes misses relevant terms outside the window. Instead, being the same as what we describe in Section 3.3.1, we adopted a temporal decay model to update the profile. For simplicity, the decay model does not consider absolute time; instead, we treat the time between updates as a time unit. Under this representation $\{p_1, p_2, ..., p_n\}$ is a sequence of posts in blog channel b and each post p_i is represented as a weighted term vector $\vec{V}_{p_i}$.

The blog channel profile vector $\vec{V}_b^1$ is initially set to $\vec{V}_{p_1}$ upon arrival of post p_1. As each new post p_i arrives, the blog channel profile vector $\vec{V}_b^{i-1}$ is updated to $\vec{V}_b^i$ as follows:

$$\vec{V}_b^i = \theta \cdot \vec{V}_b^{i-1} + (1 - \theta) \cdot \vec{V}_{p_i}$$

θ is a temporal decay factor, $0 < \theta < 1$; we choose an appropriate value for θ based on tuning within experimental datasets.

4.3.2.2 Computing Profile Similarity

A similarity metric M_{sim} determines a similarity score between two profiles. A profile is represented as a weighted term vector. The similarity of two profiles is computed in the same manner as computing the similarity between two documents. We use a variant of the state-of-the-art Okapi formula [84] to calculate similarity. The similarity between two vectors (for profiles) $\vec{V}_i$ and $\vec{V}_j$ is computed as follows:

$$M_{sim}(\vec{V}_i, \vec{V}_j) = \sum_t w_{tf}^i(t) \times w_{tf}^j(t) \times w_{idf}(t)$$

where $w_{tf}^x(t)$ is the *tf*-weight of term t in vector $\vec{V}_x$ and $w_{idf}(t)$ is the *idf*-weight. More details about calculating the *tf*-weight and the *idf*-weight are described in

Section 3.2.2.1.

4.3.2.3 Content Based Prediction (CBP)

After the profiles are built and indexed, CBP will retrieve the top K blog channels ranked by their profile similarity scores to the focal blog channel. The detail of the method is described in algorithm 4 which is labeled as *CBP*.

```
Algorithm 4 Content Based Prediction (CBP)
Input: Profile Index, the target blog channel $b$, $K$
Output: Top K blog channels
1. Query the profile index. For each blog channel $b'$ whose profile contains any common terms with the profile of $b$, compute the similarity score $Sim(b, b')$.
2. Select top $K$ blog channels with the descending order of their similarity scores with $b$.
```

4.3.3 Hybrid Prediction Method (HYBRID)

We consider a simple unsupervised learning approach that combines the predictions of *CBP* and *Bonacich-A* (the best predictor from the link based methods. HYBRID will be used to set a baseline to compare a supervised learning approach. Recall that *CBP* and *Bonacich-A* both generate a Top K ranked list. There are many methods to merge ranked lists; a popular approach is based on the Borda count. While it is a simple solution it has the drawback that it gives equal weight to all rankings. In FLP, when there are historical links, then the prediction made by *Bonacich-A* is often superior to that made by *CBP*. We develop a method *HYBRID*

Algorithm 5 Hybrid Prediction Algorithm (**HYBRID**)

```
Input: Profile Index, target blog channel $b$, LINK graph adjacency matrix $A$, $K$, $\alpha$, $\beta$, $D$
Output: Top K blog channels
1. Let $i$ denote the node of the target blog channel $b$. Compute the value of $(C(\alpha, \beta))_{i,j}$ for all other blog channels $j$ such that there is a shortest path with distance not greater than $D$ between $i$ and $j$. Create up to top $K$ blog channel ranked list $L_l$ based on the values of $(C(\alpha, \beta))_{i,j}$.
2. Use CBP to create top K blog channel ranked list $L_c$ ranked by similarity scores.
3. Get the intersection list $L_{com}$ which appear in both $L_c$ and $L_l$. The items in $L_{com}$ are ranked by the corresponding locations in $L_l$.
4. Let $L_{sep1}$ be the list of blog channels that appear $L_l$ but not in $L_c$. The order of $L_{sep1}$ is determined by the order of $L_l$. Let $L_{sep2}$ be the list of blog channels that appear $L_c$ but not in $L_l$. The order of $L_{sep2}$ is determined by the order of $L_c$.
5. Create $L_{new}$ by first appending $L_{sep1}$ to $L_{com}$ and then appending $L_{sep2}$ to it. Return the top $K$ blog channels in $L_{new}$.
```

that is inspired by the Borda count but favors the ranking of *Bonacich-A* when there are historical links.The detail of the method is described in algorithm 5 which is labeled as *HYBRID*.

4.3.4 Ranking SVM Based Prediction (RSVMP)

We apply ranking SVM [51] to rank the set of candidate blog channels for prediction, for a focal blog channel. We consider the following features for training the ranking SVM and we report on their effectiveness: (1)*FT-PROFILE*: The similarity score between the profile of the focal blog channel b and the profile of candidate blog channel b'; (2) *FT-INSIDELINKS-BONACICH*: The Bonacich score of the candidate blog channel b' with respect to the focal blog channel b using blog-channel-to-blog-channel links; (3) *FT-INSIDELINKS-COMMONNEIGHBOR*: The *CommonNeighbors* score of the candidate blog channel b' with respect to the focal blog channel b using blog-channel-to-blog-channel links; (4) *FT-EXTERNALLINKS*: The *CommonExternal* score of the candidate blog channel b' with respect to the focal blog channel b based on external links.

4.4 Experimental Evaluation

4.4.1 Evaluation Dataset and Metrics

4.4.1.1 Dataset

We use the same blogoshpere dataset as what we use for the Future Author Prediction Problem in Chapter 3. The original dataset is provided by Spinn3r.com. How we preprocessed the original dataset is described in Section 3.4.1.1. After preprocessing, the statistics describing the blog channel experiment data set is in Table 4.1.

Table 4.1: Statistics of the blog channel experiment data set

Time range	07/30/08–10/1/08
Number of blog posts	2,185,810
Number of blog channels	42,005
Average number of posts per blog channel	52.04
Number of external links (links pointing to outside of the dataset)	7,883,004
Number of bog-channel-to-blog-channel links without self references	154,218

The blog-channel-to-blog-channel links were created as follows:

- If there is a link pointing from any post in blog channel b_i to blog channel b_j or pointing to any post in blog channel b_j, we put a blog-channel-to-blog-channel link from b_i to b_j in the blog channel link graph.
- We ignored self-referential links, i.e. a post in b_i points to a post in b_i or directly to b_i.

Table 4.1 shows that there are 154,218 blog-channel-to-blog-channel links without self references within the dataset. 109,388 are post-to-blog-channel links and 44,830 are post-to-post links. On average, there are 3.67 blog-channel-to-blog-channel links for each blog channel. We further analyze the time span (TS) in days of post-to-post links. The time span reflects the number of days (d_2 - d_1) that

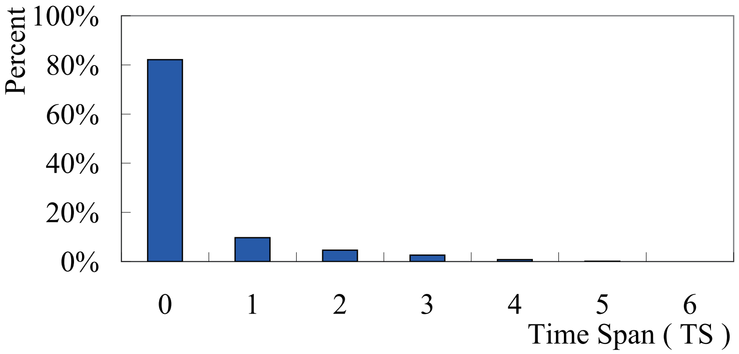


Figure 4.2: The time span (TS) for post-to-post links. The Y axis is the percentage of post-to-post links. More than 80% post-to-post links have a TS of 0 days and 8% have a TS of 1 day.

have elapsed between the day (d_1) of a post (p_1) and the day (d_2) when a post (p_2) appears that has a link to p_1. We note that the value of TS is short. More than 80% are links to posts that appeared within the previous 24 hour time interval. The largest value for TS is 6 days. Figure 4.2 shows the distribution of TS values for the entire dataset.

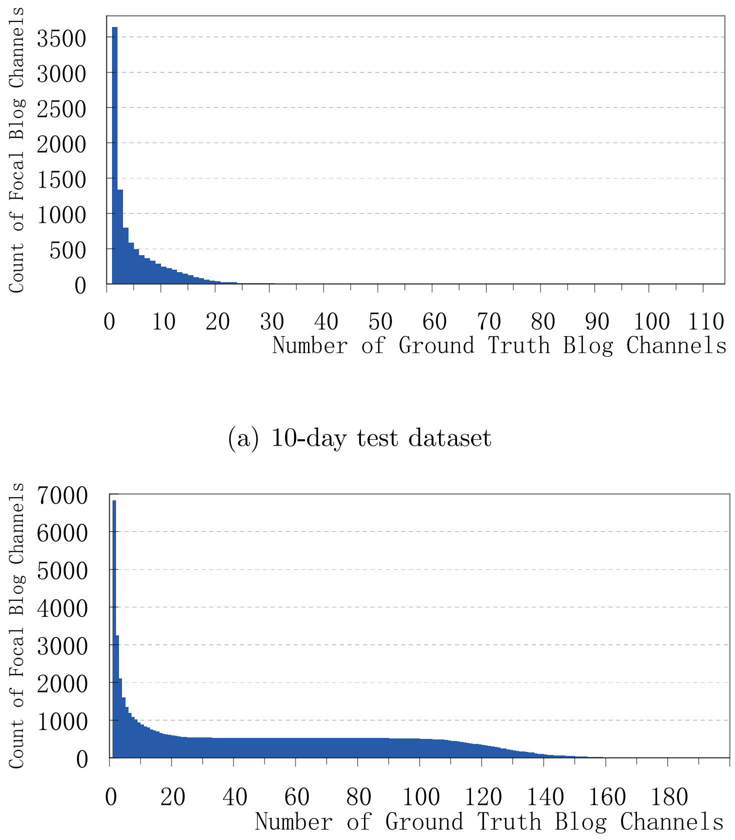
4.4.1.2 Test Datasets and Ground Truth

We created two test datasets to obtain ground truth. One test dataset included 10 days of posts from September 1 to September 10, another included 30 days of posts from September 1 to October 1. The subset of blog-channel-to-blog-channel links that are used to determine the ground truth are those links starting from a post in the test data, and pointing to a post in a focal blog channel or pointing to a focal blog channel. This created two sets of focal blog channels, S_1 and S_2. S_1 is the set of focal blog channels that contain ground truth in the 10-day test dataset, and S_2 is the set of all blog channels that contain ground truth in the 30-day test dataset. S_1 includes 3636 focal blog channels while S_2 includes 6831.

Figure 4.3 reports on the distribution of the number of ground truth blog channels for the focal blog channels in S_1 and S_2. A small number of the focal blog channels have a large number of ground truth blog channels while a large number have a small number of ground truth blog channels. This is consistent with a power law distribution. The focal blog channels in S_1 have an average of 2.7 ground truth blog channels. The vast majority of 99.3% have fewer than 20 ground truth blog channels while 21.8% have at least 2. The focal blog channels in S_2 have an average of 12.1 ground truth blog channels. 91.7% have fewer than 20 ground truth blog channels while 31.0% have at least 2.

4.4.1.3 Training Data

The training data was obtained from July 30 to August 31. All historical posts and blog-channel-to-blog-channel links in this 31 day interval were used by the link based methods and *CBP* and *HYBRID* in a straightforward manner. For *RSVMP*, we selected a training dataset of 10 days starting from August 22. The training focal blog channels T_1 are those blog channels that contain ground truth within the 10 day training dataset. The features of the blog channels that were used to produce the training pairs for each training focal blog channel were collected in the preceding time interval from July 30 to August 21. We reiterate that there was



(a) 10-day test dataset

(b) 30-day test dataset

Figure 4.3: The distribution of the number of ground truth of the focal blog channels. The value on Y axes is the count of the focal blog channels whose ground truth contains at least x blog channels, where x is the value on the X axis.

no overlap between the training and testing time interval. Similarly there was no overlap in the time interval for feature collection and to obtain the training ground truth for *RSVMP*.

4.4.1.4 Metrics and Parameters

Precision, recall and the F measure are computed on unordered sets of documents. Mean Average Precision (MAP) is widely used for evaluating ranking

Table 4.2: Parameters for experiments

Parameter	Value	Description
K	20	Top K prediction
α	0.002	The Bonacich attenuation factor for indirect links
β	1.0	The Bonacich attenuation factor for direct links
D	10	The threshold of the minimum path length to the focal blog channel
θ	0.8	The temporal decay factor for building blog channel profiles

methods. It provides a single-figure measure of quality across recall levels. MAP has been shown to have good discrimination and stability [69] and we report the values of MAP. We use the Wilcoxon signed-rank test [106] to determine statistical significance of results.

Table 4.2 describes the parameters and their values for our experiments. We selected $K = 20$ for evaluation since more than 90% of the focal blog channels have no more than 20 ground truth blog channels. The values of α, β, θ were tuned from the training data. We selected $D = 10$ for *Bonacich-A*. For this dataset, there was no benefit for values of D greater than 10, while there was a significant computational overhead.

4.4.2 Experimental Results

4.4.2.1 Baseline for the 3 methods

Figure 4.4 reports on MAP for all of the methods on the two test datasets. *CBP* which utilizes content for prediction has the lowest prediction accuracy. All other methods which utilize historical links for prediction have higher accuracy than *CBP*. This shows that links are the most significant prediction feature for FLP. Among the link based methods, *Bonacich-A* dominates *Jaccard* and *CommonNeighbors* and *CommonExternal*. Recall that *CommonExternal* exploits external links for prediction. Its accuracy is similar to *CommonNeighbors* and this demonstrates that external links are also a good feature for prediction. *HYBRID* can benefit from combining links and content based features. Finally, the supervised learning method *RSVMP* dominates all of other methods.

While the values for MAP in Figure 4.4 may appear to be low values, we note that for comparable social media tasks, e.g., the TREC blog distillation task [77], the reported MAP values are also often quite low (in the range of 0.10-0.30). The Wilcoxon signed-rank test shows that *RSVMP* significantly outperforms *HYBRID* and *HYBRID* significantly outperforms *Bonacich-A*, with p far less than 0.01.

For each prediction method, the MAP value for the 10-day test dataset is higher than the 30-day test dataset. This is a surprising and interesting result. This reflects our argument that the blogosphere evolves in many ways. Both the topical interests of the bloggers and their continuing interest in following bloggers changes over time. For example, we observe that 14.6% of the focal blog channels

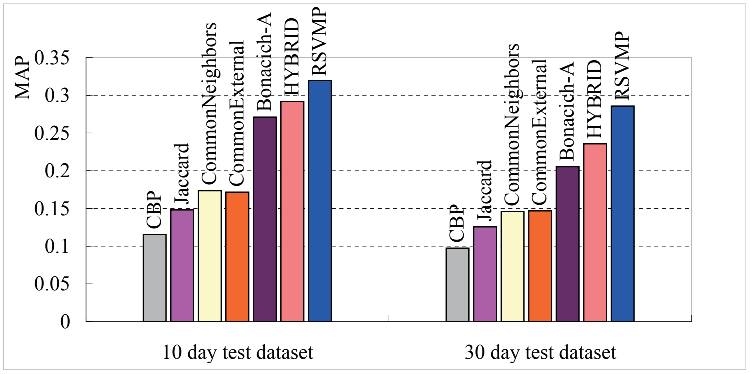


Figure 4.4: The performance of the prediction methods.

in S_1 which have ground truth in the 10-day test dataset do not have historical blog-channel-to-blog-channel links in the training dataset. In comparison, 27.0% of the focal blog channels in S_2 which have ground truth in the 30-day test dataset do not have historical blog-channel-to-blog-channel links in the training dataset. This is consistent since the interval of the training data may be quite distant in time from some of the events, i.e., the posts that contain links in S_2. In other words, the significance of the historical links reduces or expires over time as the interest of their followers changes over time.

4.4.2.2 Subset with no historical blog-channel-to-blog-channel links

Figure 4.5 reports MAP values for all of the methods on the subset of the focal blog channels without historical bog-channel-to-blog-channel links. The methods *Jaccard*, *CommonNeighbors* and *Bonacich-A* which only use bog-channel-to-blog-channel links for prediction can make no prediction and have a 0.0 MAP value. *HYBRID* has the same MAP value as *CBP*. *CommonExternal* outperforms *CBP*,

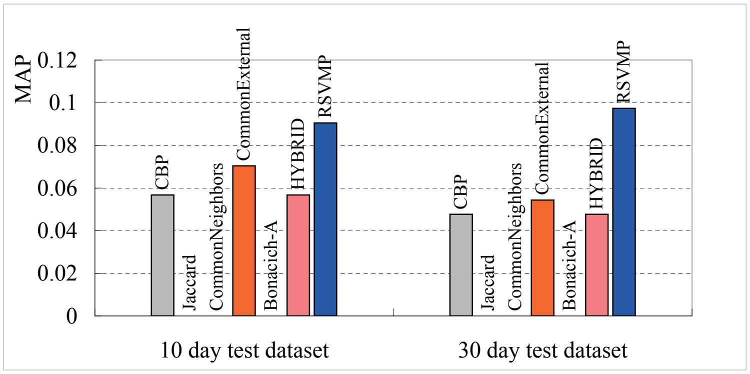


Figure 4.5: The performance of the prediction methods on the subset of the focal blog channels with 0 blog-channel-to-blog-channel historical links.

indicating that the external links are more significant predictors compared to the blog channel profile alone. As expected, *RSVMP* dominates all methods. While the prediction accuracy is not very high (all have MAP values of less than 0.1), we note that this is a very challenging prediction task for the noisy blogosphere with zero historical blog-channel-to-blog-channel links. Thus, despite the low MAP value, this experiment demonstrates that *RSVMP* can perform surprisingly well and can exploit content and external links for a very difficult scenario of FLP.

4.4.2.3 Feature analysis for RSVMP

Figure 4.6 reports MAP values for *RSVMP* method for different features. We consider two groups of features. One group of features is based on blog-channel-to-blog-channel links, i.e. network features. There are two features in this group: *FT-INSIDELINKS-BONACICH*, and *FT-INSIDELINKS-COMMONNEIGHBOR*. The other group of features are content based and include *FT-PROFILE* and *FT-*

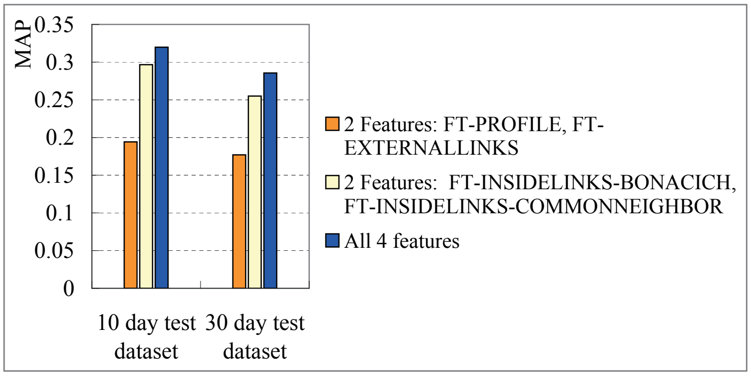


Figure 4.6: The performance of the *RSVMP* by applying different features.

EXTERNALLINKS. Note that while *FT-EXTERNALLINKS* represents links, the value of these links are the content of the referenced pages.

Figure 4.6 shows that *RSVMP* has better performance when applying the group of network based features alone, in comparison to applying the content based features alone. Also as expected, *RSVMP* has the best performance when it combines both groups of features.

Chapter 5

Prediction in a Hybrid Network from Microblogs

5.1 Introduction

Microblogs such as Twitter support a rich variety of user interactions. One can follow a user and read her tweets. One can search for keywords or hashtags or follow trending tweets. A user can initiate a new topic by creating a new tweet or hashtag, often including a url in the tweet to refer to more detailed articles. One can interact with another user by mentioning them. One can also participate in the diffusion of a topic by retweeting. Retweets and mentions have been identified as an important proxy for influence [21]. All of these interactions create a dynamic and rich social network for diffusion of information and to establish the influence of a user. There has been much work on modeling diffusion and influence in a variety of networks and media. Our objective is to develop a model that can capture the richness and complexity of microblogs. We make *predictions about the future at the level of an individual user*, i.e., given a focal user, we want to predict the other users who will interact with her.

One motivation for this research is to support monitoring for personalized and interactive brand management. A brand manager has the objective of monitoring conversations about a brand, to track relevant topics and sentiment, and to identify potentially negative conversations. While aggregate statistics, e.g., a trending topic about the brand, or an increase in negative sentiment, is important, social media also provides a platform for personalized and interactive brand management. Our objective is to use prediction models at the individal level to make personalized recommendations of influential and relevant and diverse users. Suppose a brand manager knows which user u is relevant and is likely to talk about her brand, either in a positive or negative way. It will be useful if she could determine if u is influential, and other users who will be influenced by u. With this knowledge, she could perhaps take a proactive action such as engaging in a conversation with u and those who may be influenced by u. Diverse recommendations may target a user v who has not previously tweeted about the brand but who has several friends who are interested in the brand and have retweeted relevant tweets.

One advantage of microblogs is that it is simple to monitor the streams due to the brevity of microblogs; hashtags and urls enhance the stream with richer content and links. More important, diffusion can be easily monitored through retweets and mentions. However, the popularity of social media creates a deluge of noisy and irrelevant data streams. A typical brand manager may be overwhelmed by the amount of users and information that she would have to monitor. Individual influence analysis could then help facilitate personalized recommendation by effectively filtering uninteresting information and delivering high-quality personalized recommendations.

One observation for microblogs is that influence is not limited to the immediate neighborhood, i.e., influence can spread outside the friendship network of microblogs. One can retweet or mention users who are not one's friends. For example, in our experimental dataset (described in Section 5.4), more than 40% of the mentions are from outside the follower network.

More important, Twitter is an example of a *hybrid network* composed of multiple networks. There is an explicit Follower network. Retweet actions and mention actions also reflect key relationships between users and hence Retweet and Mention networks can be constructed. Finally, in other media, influence will only result in the evolution of a single network, typically the equivalent of the Follower network of Twitter. However, in microblogs, we expect all three networks to evolve as a result of the influence of the user. For example, if v is retweeted a lot, she may attract additional followers, and that in turn may lead to even more retweets and mentions from the followers of the users who recently joined the Follower network of v. This is an example of the Retweet network causing an evolution of the Follower network, which in turn results in an evolution of the Retweet and Mention networks, respectively.

To summarize, microblogs exhibit complex user interactions over a hybrid network. Influence is not limited to the immediate followers and it can be measured through the characteristics of the three networks. Further, the impact of influence may result in the evolution of all three networks. Our work has the following distinguishing features:

- We develop an accurate prediction model at the individual level. We want to understand who will be influenced by a particular focal user.

- Prediction of future retweets and future mentions is unique to our work. Our objective is personalized and interactive recommendations for brand management.

- We consider a hybrid network. Unlike traditional link prediction [64] over a homogeneous network, our challenge is to model an evolving hybrid network and to exploit this hybrid network to improve prediction accuracy.

Our approach can be summarized as follows:

- We define a hybrid network made up of a Follower network, a Retweet network, and a Mention network. We define two link prediction problems: the future retweet prediction and the future mention prediction.
- We define a potential function over the hybrid network that reflects the likelihood of a candidate user having a specific type of link in the future to a focal user.
- We formalize this future link prediction problem in the hybrid network as an optimization problem using the maximum likelihood principle.
- We propose heuristic solutions to approximate the optimization problem and reduce its computational complexity.
- We perform an extensive evaluation over a microblog network and a stream of tweets from Twitter. Our solutions outperform baseline methods which only consider one network or naively utilize the hybrid network.

5.2 Problem Formulation

5.2.1 Problem Definition

Definition 3 ***Future Retweet Prediction:*** *Given a focal microblog user u at a specific time point T and a time interval ΔT, identify K microblog users $S^u_{T,\Delta T}$ who will retweet one (or more) future tweet(s) of user u in the interval $(T, T+\Delta T]$.*

Definition 4 ***Future Mention Prediction:*** *Given a microblog user u at a specific time point T and a time interval ΔT, identify K microblog users $S^u_{T,\Delta T}$ who will mention microblog user u one (or more) times in the interval $(T, T+\Delta T]$.*

5.2.2 Prediction Model

Our objective is to exploit historical knowledge and the corresponding hybrid network to accurately predict future links. Let $G^1, \cdots, G^M$ represent the M relationship networks constructed using history; for Twitter M=3 and there are Follower, Mention and Retweet networks. The corresponding relationship networks in the future time period are denoted by $Y^1, \cdots, Y^M$. Our objective is to infer an optimal composite network H^c from $G^1, \cdots, G^M$ to predict each Y^c, $1 \leq c \leq M$. To be optimal, the hybrid network for each Y^c should be a customized network H^c.

Let $G^m_{i,j}$ represent the weight associated with the edge from node j to node i in some network G^m. Let $H^c_{i,j}$ represent the weight associated with the edge from j to i in the hybrid network H^c.

We define the hybrid network H^c for each Y^c as follows:

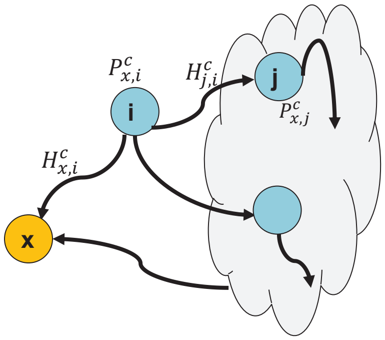


Figure 5.1: Prediction model. $P^c_{x,i}$ or $P^c_{x,j}$ is the potential of i or j having a type c link to x in the future. $H^c_{x,i}$ and $H^c_{x,j}$ are the weights of the hybrid network edges.

$$H^c_{i,j} = \sum_m \omega_m G^m_{i,j} \quad \text{where } \forall m, \omega_m \geq 0. \tag{5.1}$$

A potential function $P^c_{x,i}$ defined over each hybrid network H^c reflects the likelihood of a candidate node i having a link of type c in the future to a focal node x. We define this potential function $P^c_{x,i}$ as follows:

$$P^c_{x,i} = \beta H^c_{x,i} + \alpha \sum_j P^c_{x,j} H^c_{j,i} \quad \text{where } \alpha \geq 0, \beta \geq 0. \tag{5.2}$$

For a focal node x, we consider several factors contributing to the potential from node i. The first factor is the weight of the hybrid network edge from node i to node x. The second factor is the potential of the neighbors j of node i to focal node x. The third factor is the weight of the hybrid network edges from node i to its neighbors j. Figure 5.1 is the visualization of these factors towards the potential function.

We can finally define a conditional probability to determine whether node i

will have a type c link to node x in the future, based on the potential of node i. Similar to [79], we adopt an exponential probability distribution to determine this function. We define our conditional probability as follows:

$$Prob(Y^c_{x,i} > 0 \mid G^m, 1 \leq m \leq M) = 1 - \exp(-s - P^c_{x,i}) \tag{5.3}$$

$Y^c_{x,i}$ is the weight of a future type c edge from node i to node x. If the value is greater than zero, it means that such a future edge exists. In the right side of the equation, s is a parameter.

The exponential function of $f(x) = exp(-x)$ has the monotonic and concave properties and matches the recent research [20] which suggests that the probability of adoption increases at a decreasing rate with increasing external network signals [79].

5.2.3 Solution Approach

In equation (5.2), $H^c_{x,i}$ is the weight of the hybrid network edge from node i to node x, $H^c_{j,i}$ is the weight of the hybrid network edge from node i to node j, α and β are two parameters. Here we have potential values on both sides of the equation. By solving a group of these equations, we can represent the potential values by other factors and parameters rather than having potential variables on the right side. By applying equation (5.1) to equation (5.2), we have the following formula:

$$P^c_{x,i} = \beta \sum_m \left(\omega_m \cdot G^m_{x,i}\right) + \alpha \sum_j \left\{ P^c_{x,j} \sum_m \left(\omega_m \cdot G^m_{j,i}\right) \right\} \tag{5.4}$$

By recursively replacing $P^c_{x,j}$ with its expression in equation (5.4), we have the following expression of $P^c_{x,i}$.

$$\begin{aligned} P^c_{x,i} =& \beta \sum_m \left(\omega_m \cdot G^m_{x,i}\right) \\ & + \beta\alpha \sum_j \left\{ \sum_m \left(\omega_m \cdot G^m_{x,j}\right) \sum_m \left(\omega_m \cdot G^m_{j,i}\right) \right\} \\ & + \beta\alpha^2 \sum_j \sum_{j_1} \left\{ \sum_m \left(\omega_m G^m_{x,j_1}\right) \sum_m \left(\omega_m G^m_{j_1,j}\right) \right. \\ & \left. \sum_m \left(\omega_m G^m_{j,i}\right) \right\} \\ & + \cdots\cdots \end{aligned} \tag{5.5}$$

In equation (5.5), the first factor is the impact of the paths of length 1 to the potential, and the second factor is the impact of the paths of length 2 to the potential, and the third factor is the impact of the paths of length 3 to the potential. It contains infinite factors. The expression turns out to be a generalization of the Bonacich metric [15]. The original Bonacich metric only considers one type of path; here we have hybrid paths.

To train the model, we need to estimate the optimal values for α, β, $\omega_1, \cdots, \omega_M$, and s. We can formalize it as an optimization problem by maximizing the product of all conditional likelihood expressions. We maximize the contribution of positive values of the expression and we minimize the negative values. Using logarithmic operations, the product reduces to a summation. Given training data to construct the historical graphs $\{G^m, 1 \leq m \leq M\}$ we can compute the following probabilities to predict the graph Y^c:

$$\underset{s,\alpha,\beta,\omega_1,\cdots,\omega_M}{arg\ \ max}\ f_c(s,\alpha,\beta,\omega_1,\cdots,\omega_M),$$
$$\text{Subject to:}\quad s \geq 0, \alpha \geq 0, \beta \geq 0, \forall m, \omega_m \geq 0 \tag{5.6}$$

where:

$$\begin{aligned} f_c(s,&\alpha,\beta,\omega_1,\cdots,\omega_M) \\ &= \log\left\{\prod_x \prod_{i:Y^c_{x,i}>0} \Pr ob(Y^c_{x,i} > 0|G^m, 1 \leq m \leq M)\right. \\ &\quad \left.\prod_{i:Y^c_{x,i}=0}\left[1 - \Pr ob(Y^c_{x,i} > 0|G^m, 1 \leq m \leq M)\right]\right\} \\ &= \sum_x\left\{\sum_{i:Y^c_{x,i}>0} \log(1 - \exp(-s - P^c_{x,i}))\right. \\ &\quad \left. - \sum_{i:Y^c_{x,i}=0}(s + P^c_{x,i})\right\} \end{aligned} \tag{5.7}$$

Given a dataset with n nodes, there will be n^2 items in equation (5.7), and each item contains a different potential $P^c_{x,i}$ which is expensive to compute according to equation (5.5). For large n, any strategy to solve the optimization problem (5.6) will be prohibitively expensive. We explore approximate solutions in the next section.

5.2.4 Representing Microblog Networks

Figure 1.2 illustrates user interactions in Twitter. We define the following adjacency matrices for the Retweet, Mention and Follower networks:

- M: Mention network adjacency matrix; the value of $M_{i,j}$ is the number of mentions from user j of user i.

- R: Retweet network adjacency matrix; the value of $R_{i,j}$ is the number of retweets by user j of tweets from user i.

- F: Follower network adjacency matrix; $F_{i,j}$ is equal to 1 if user j is a follower of user i; else $F_{i,j}$ is equal to 0.

We define F^* to be an adjusted follower network adjacency matrix reflecting the popularity factor of each follower. $F^*_{i,j}$ is a weighted value if user j is a follower of user i; otherwise $F_{i,j}$ is equal to 0. If user j is a follower of user i, the weighted value of $F^*_{i,j}$ is affected by the number of friends of that user j. We calculate $F^*_{i,j}$ as follows:

$$F^*_{i,j} = \frac{\overline{D}}{D_j}$$

$\overline{D}$ is the average of number of friends over all users; D_j is the number of friends of user j. The intuition is that if a user j has a lot of friends, then her attention will be divided among those friends, and she will pay less attention to user u. Consequently, user u has a lower influence on user j, if j has a lot of friends.

5.3 Approximate Solutions

5.3.1 Intuition for Approximation

We consider two alternative approaches to approximate the optimization problem presented in equation (5.6) to determine the optimal hybrid network(s) H^c of equation (5.1).

The first approach is to approximate the hybrid network H^c. In this case equation (5.1) can be expressed as follows:

$$H^c = r \cdot R + m \cdot M + f \cdot F^*$$

where r, m, and f are the weights associated with the Retweet, Mention and Follower adjacency matrices, R, M and F^*, respectively. Instead of solving the optimization problem of equation (5.6) to get optimal values of $\omega_1, \omega_2, \cdots$, we will apply some heuristics method to approximate the values of r, m, and f. We then calculate the score matrix P using equation (5.5). P can be used for prediction. Method **WT-COM-BON** is based on this approach.

A second approach bypasses the optimization problem of equation (5.6), and the optimal hybrid network(s) H^c of equation (5.1). This approach will directly consider different types of paths that combine edges from the Retweet, Mention and Follower networks. For example, Figure 5.2 shows the relationship between three nodes x, y, z. There are three types of links, labeled as a, b, c. Suppose user z follows user y, and user y mentions user x. Then there is a path from z to x with length of two steps, by first a following action and then a mentioning action. There may be different types of paths possibly with different lengths between two users. Different types of paths should be given different credit for prediction. We denote the adjacency matrices for types of links a, b, c as A, B, C respectively. The paths whose path length is 1 is represented by the adjacency matrix, A, B, C. The matrix AB which multiplies A and B represents the weight of path $\longrightarrow^{b}\longrightarrow^{a}$ between each pair of nodes. Our target is to use some function g' to generate a score matrix for

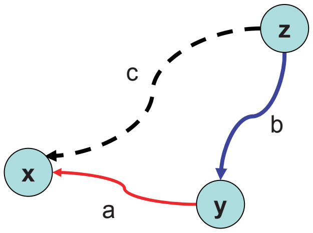


Figure 5.2: Hybrid path. x, y, z are three nodes and a, b, c are the labels for three types of links.

an approximate potential function P' exploiting such paths as follows:

$$P' = g'(R, M, F^*, RM, MR, RF^*, F^*R, \cdots)$$

P' is then used for prediction by **MIX-PATH**.

We use the following intuition to determine the approximate values for r, m, f and g': Suppose the adjacency matrix A is correlated with C. Then, a higher weight associated with an edge of type a, from y to x, should result in an increase in the probability of a type c edge from y to x. Similarly, suppose AB is correlated with C. Then, a higher weight associated with the path $\longrightarrow^{b}\longrightarrow^{a}$ from z to x, should result in an increase in the probability of a type c edge from z to x.

5.3.2 Factors for Approximation

For the first approximate approach, where we want to first create a composite network, $H^c = r \cdot R + m \cdot M + f \cdot F^*$, we consider two factors for the weights r, m, f. We first scale the matrices so that no matrix can dominate the others. We then use the ground truth from the training data to calibrate the influence of each network R, M, F^* with respect to retweet prediction and mention prediction. These two factors can also be applied to the second approximate approach where we use different types of paths directly. We will explain it later when we describe that approach.

5.3.2.1 Scale Factor

The scale of each of the matrices may be different, e.g., the distribution of the values. This can result in one matrix dominate another.

We define a scale factor γ. Given a matrix B, then $\gamma(A, B)$ will scale matrix A with respect to B. For retweet prediction, R is the matrix that has the greatest influence and is used as a standard. We use the summation of the weights (values) of R, $\sum_{i,j} R_{i,j}$, to determine γ for retweet prediction. The scale factors for the three different networks with respect to retweet prediction are formally calculated as follows:

$$\gamma(R, R) = 1,\ \gamma(M, R) = \frac{\sum_{i,j} R_{i,j}}{\sum_{i,j} M_{i,j}},\ \gamma(F^*, R) = \frac{\sum_{i,j} R_{i,j}}{\sum_{i,j} F^*_{i,j}}$$

The meaning of using retweet network as standard is that there are some total number of retweets among the users in the training period. Scaling other network to this standard is to mimic the retweet relationship but with somewhat different distribution.

Similarly, the scale factors for the three different networks with respect to mention prediction are formally calculated as follows:

$$\gamma(R,M) = \frac{\sum_{i,j} M_{i,j}}{\sum_{i,j} R_{i,j}},\ \gamma(M,M) = 1,\ \gamma(F^*,M) = \frac{\sum_{i,j} M_{i,j}}{\sum_{i,j} F^*_{i,j}}$$

5.3.2.2 Penalty Factor

We define penalty factors to lessen the weight of the networks which have lower prediction ability. Intuitively, for retweet prediction, the matrix R (which matches the retweet ground truth from training data) is the most important. If another network M or F^* deviates from R then a penalty factor must be imposed. We assign the penalty factors to other networks according to their correlation with the ground truth network in the training data, i.e., the network in the history with the same type of link to predict in the future.

We use average Spearman's rank correlation coefficient as the metric of how two adjacency matrices are correlated. A high correlation means a low deviation. For two rank sets X, Y, suppose x_i and y_i are the ranks of the values of X_i and Y_i in X and Y respectively, and the number of elements in X and Y are both n, Spearman's rank correlation coefficient is calculated as:

$$\rho(X,Y) = 1 - \frac{6\sum(x_i - y_i)}{n(n^2-1)}$$

The closer ρ is to +1 or -1, the stronger the correlation is. A perfect positive correlation will have a ρ value +1 and a perfect negative correlation will have a ρ value -1. We only consider the positive correlation. The penalty factor should be related to the correlation. The lower the correlation, the penalty will be stronger, i.e., the penalty factor should be lower. We define a penalty factor to be a function

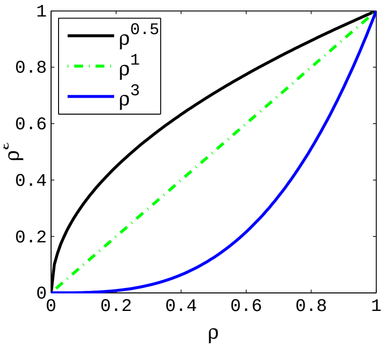


Figure 5.3: Function ρ^{ε}.

with respect to ρ as ρ^{ε}. Figure 5.3 shows the curves of ρ^{ε} with respect to different ε values. When ρ is in the range of (0,+1], ρ^{ε} will also be in the range of (0,+1]. When ε is greater than 1, ρ^{ε} has lower value than ρ. The bigger value of ε, the stronger penalty is applied with the same correlation value.

The penalty factors for the three different networks with respect to retweet prediction are formally calculated as follows:

$$\begin{aligned} &\varphi(R,R) = 1, \\ &\varphi(M,R) = \left\{\frac{1}{N}\sum_{i}\rho(M_i,R_i)\right\}^{\varepsilon}, \\ &\varphi(F^*,R) = \left\{\frac{1}{N}\sum_{i}\rho(F_i^*,R_i)\right\}^{\varepsilon} \end{aligned} \tag{5.8}$$

Where $\rho(M_i,R_i)$ is the Spearman's rank correlation of ith row of M and ith row of R, and $\rho(F_i^*,R_i)$ is the Spearman's rank correlation of ith row of F^* and ith row of R, and N is the number of rows.

Similarly, the penalty factors for the three different networks with respect to mention prediction are formally calculated as follows:

$$
\begin{aligned}
&\varphi(R, M) = \Big\{\frac{1}{N}\sum_i \rho(R_i, M_i)\Big\}^{\varepsilon}, \\
&\varphi(M, M) = 1, \\
&\varphi(F^*, M) = \Big\{\frac{1}{N}\sum_i \rho(F_i^*, M_i)\Big\}^{\varepsilon}
\end{aligned}
\tag{5.9}
$$

5.3.2.3 Merging Parameters

The merging parameters r, m, f to create the composite network are composed by scale factors and penalty factors. For retweet prediction,

$$
\begin{aligned}
&r = \gamma(R, R) \cdot \varphi(R, R), \\
&m = \gamma(M, R) \cdot \varphi(M, R), \\
&f = \gamma(F^*, R) \cdot \varphi(F^*, R)
\end{aligned}
\tag{5.10}
$$

For mention prediction,

$$
\begin{aligned}
&r = \gamma(R, M) \cdot \varphi(R, M), \\
&m = \gamma(M, M) \cdot \varphi(M, M), \\
&f = \gamma(F^*, M) \cdot \varphi(F^*, M)
\end{aligned}
\tag{5.11}
$$

The composite network H^c is created by merging the hybrid networks as

$$H^c = r \cdot R + m \cdot M + f \cdot F^*$$

5.3.3 Approach Using the Composite Network: WT-COM-BON

After we have the composite network, for the focal user x, we can calculate the potential values of all candidate users according to equation (5.5), with respect to the likelihood of having type c link to x in the future. Since we already know the weights of different networks, we can simplify the equation (5.5) by using equation (5.1) as:

$$P^c_{x,i} = \beta H^c_{x,i} + \beta\alpha(H^c \cdot H^c)_{x,i} + \beta\alpha^2(H^c \cdot H^c \cdot H^c)_{x,i} + \cdots\cdots \tag{5.12}$$

The formula (5.12) has the same format as the *Bonacich centrality* [15]. *Bonacich centrality* is a generalization of Katz's metric [52]. *Katz* measures the status of a node by the total number of paths linking it to other nodes in the graph; an exponential discount is used as the path length increases. *Bonacich centrality* also reflects the total number of paths originating from a node and uses an attenuation factor α to discount indirect links and β to discount direct links. *Katz* has been shown to be one of the best topological methods in [64] and *Bonacich* metric was shown to outperform metrics like *Jaccard* and *CommonNeighbors* in [110].

Let A be the adjacency matrix. Recall that each node in the link graph is a user. Suppose T is the time when the prediction is to be made. Thus, we consider all historical links prior to T. We set the value of the element $A_{i,j}$ in the adjacency matrix A is the number of direct links of some type pointing from node j to node i that exist in history before time T. Then, the value of $(A^n)_{i,j}$ is equal to the number of paths of length n from node j to node i in history. Bonacich centrality is computed as follows:

$$\begin{aligned} C(\alpha,\beta) &= (\beta A + \beta\alpha A \cdot A + ... + \beta\alpha^n A^{(n+1)}...) \\ &= \beta A(1-\alpha A)^{(-1)} \end{aligned}$$

This equation holds while $\alpha < 1/\mu$, where μ is the largest characteristic root of A [34]. For $\alpha = \beta$, this measure reduces to the Katz score.

Unlike A in the above equation, where the value of each element is the number of links from a node to another node, for matrix H^c of a composite network, the value of each element is a real number which represents the weight of the edge from a node to another node. Then, rather than being equal to the number of paths of length n from node j to node i, the value of $(H^n)_{i,j}$ can represent the weight of the path of length n from node j to node i. Let P be the matrix of all potential values $\{P^c_{x,i}\}$. According to formula (5.12), we can calculate the score matrix P as following:

$$\begin{aligned} P &= (\beta H^1 + \beta\alpha H^2 + ... + \beta\alpha^n H^{(n+1)}...) \\ &= \beta H(1-\alpha H)^{(-1)} \end{aligned} \tag{5.13}$$

In this equation, we refer to H^c as H since we use the superscript n in H^n to refer to the power matrix expression for the matrix representation of each H^c. Then P can be used for prediction, and we label this method as **WT-COM-BON**.

We also consider two related baseline methods. One baseline method is labeled as **BON**. It computes the Bonacich metric over the Retweet network and the Mention network, R and M, for retweet prediction and mention prediction respectively. The other baseline method is labeled as **UNW-COM**. It considers the unweighted union of the edges of the Retweet, Mention and Follower networks, R, M and F^*, and computes the Bonacich metric.

5.3.4 Approach Based on the Different Types of Paths: MIX-PATH

If we rearrange equation (5.5), we can have the following expression for the potential variable $P^c_{x,i}$:

$$\begin{aligned} P^c_{x,i} =& \sum_{a_1} \left\{ \theta_{a_1} \cdot G^{a_1} \right\}_{x,i} \\ &+ \sum_{a_1} \sum_{a_2} \left\{ \theta_{a_1,a_2} \cdot G^{a_1} \cdot G^{a_2} \right\}_{x,i} \\ &+ \sum_{a_1} \sum_{a_2} \sum_{a_3} \left\{ \theta_{a_1,a_2,a_3} \cdot G^{a_1} \cdot G^{a_2} \cdot G^{a_3} \right\}_{x,i} \\ &+ \cdots\cdots \end{aligned} \tag{5.14}$$

In formula (5.14), θ_{a_1}, θ_{a_1,a_2}, and θ_{a_1,a_2,a_3} are the parameters which can be expressed by the parameters α, β, ω_{a_1}, ω_{a_2}, ω_{a_3} in equation (5.5); G^{a_1}, G^{a_2} and G^{a_3} are the different network adjacency matrices.

Let P be the matrix of all potential values $\{P^c_{x,i}\}$. From formula (5.14), we can calculate the score matrix P as following:

$$
\begin{aligned}
P = & \sum_{a_1} \left\{ \theta_{a_1} \cdot G^{a_1} \right\} \\
& + \sum_{a_1} \sum_{a_2} \left\{ \theta_{a_1,a_2} \cdot G^{a_1} \cdot G^{a_2} \right\} \\
& + \sum_{a_1} \sum_{a_2} \sum_{a_3} \left\{ \theta_{a_1,a_2,a_3} \cdot G^{a_1} \cdot G^{a_2} \cdot G^{a_3} \right\} \\
& + \cdots\cdots
\end{aligned} \tag{5.15}
$$

We need to estimate the parameters θ_{a_1}, θ_{a_1,a_2}, and θ_{a_1,a_2,a_3}, $\cdots$, to estimate the score matrix P for prediction. For this purpose, let's look at the general example in Figure 5.2 again, where there are three nodes x, y, z and three types of links a, b, c. The matrices A, B, C are the adjacency matrices for the three types of links a, b, c. The matrix AB represents the weight of path $\longrightarrow^{b}\longrightarrow^{a}$ between each pair of nodes. Suppose AB is correlated with C. The bigger weight of path $\longrightarrow^{b}\longrightarrow^{a}$ from z to x, the possibility of a type c link from z to x will be higher and with bigger weight.

Similar to the approach based on the composite network, we can use average Spearman's rank correlation coefficient between AB and C to represent their correlation, denoted as $\varphi(AB, C)$. The scale factor between AB with respect to C is calculated as:

$$\gamma(AB, C) = \frac{\sum_{i,j} C_{i,j}}{\sum_{i,j} (AB)_{i,j}}$$

Then the weight or likelihood of type c link from z to x could be somewhat proportional to $\gamma(AB, C) \cdot \varphi(AB, C) \cdot (AB)_{x,z}$.

Inspired by this intuition, to predict retweet/mention, we can consider hybrid paths as features. The credit of each feature is decided by its scale factor, correlation factor, and its path length factor. We use $\ell(X)$ to denote the path length factor of X. The path length factor is related to the parameter α in equation (5.5). In equation (5.5), for a path length with one step longer, one more α factor will be imposed. The value of path length factor is only decided by the length of the path, and is unrelated to the types of links it includes. For example, $\ell(AB) = \ell(AC) = \ell(BC) \neq \ell(A)$.

Suppose X is any of R, M, F^*, RM, MR, RF^*, F^*R, F^*M, MF^*, RR, MM, F^*F^*, $\cdots$, and C is either M or R. The weight for X with respect to C is calculated as:

$$w(X|C) = \ell(X) \cdot \gamma(X, C) \cdot \varphi(X, C) \tag{5.16}$$

Then the following score matrix is created for prediction:

$$P' = \sum_{X} \{w(X|C) \cdot X\} \tag{5.17}$$

Formula (5.17) is an approximation of Formula (5.15). We can use the matrix P' directly for prediction, and we label this method as **MIX-PATH**.

Although the `MIX-PATH` method is also deducted from our prediction model with linear combination of different networks, our estimation of its parameters would more likely to have nonlinear characteristics. For the linear combination, from equation (5.5) and equation (5.14), θ_{a_1,a_2} which is the weight of $G^{a_1}G^{a_2}$ should be equal to θ_{a_2,a_1} which is the weight of $G^{a_2}G^{a_1}$. However, $\gamma(G^{a_1}G^{a_2}, C)$ and $\varphi(G^{a_1}G^{a_2}, C)$ are very unlikely to be equal to $\gamma(G^{a_2}G^{a_1}, C)$ and $\varphi(G^{a_2}G^{a_1}, C)$, which leads to

$w(G^{a_1}G^{a_2}|C) \neq w(G^{a_2}G^{a_1}|C)$ based on equation (5.16). It means that the estimation of the weights for $G^{a_1}G^{a_2}$ and $G^{a_2}G^{a_1}$ would very likely to be different. This is reasonable since the sequence of different links in a path is also important with respect to the prediction of the ground truth. So the approximate approach `MIX-PATH` has implicitly combined different networks nonlinearly. This is the advantage of `MIX-PATH` compared with `WT-COM-BON`. The disadvantage is that there are more parameters to estimate for `MIX-PATH` and the calculation would be potentially more expensive. In practice, for `MIX-PATH` we can only pick up the paths with some limited length, but for `WT-COM-BON`, with the convenience of equation (5.13), all paths of the composite network can be included.

5.3.5 Supervised Methods for Incorporating Content Features

Content features including noun phrases and named entities, as well as sentiment, have been used successfully for link prediction [62, 110]. Given the short message length of microblogs, we focus on *hashtags* and *urls* which may signal richer content. Each user is associated with a bag of words that includes all the keywords, i.e., (shared) hashtags or urls, that occur in their tweets. The distribution of these keywords across all tweets are used to compute the term frequency (TF) and inverse document frequency (IDF) of each keyword. We then compute the bag of words similarity for pairs of users.

Based on the shared URLs and shared hashtags, we have two extra simple prediction methods, `URL-RANK` and `HTG-RANK`. The method **URL-RANK** makes prediction according to the similarity of the historical URLs, and the method **HTG-RANK** makes prediction according to the similarity of the historical hashtags. We also consider a baseline method PROLIFIC. For each focal user, the baseline method **PROLIFIC** ranks other users by their retweets (for retweet prediction) or by their mentions (for mention prediction). A user who retweets (or mentions) more will be ranked higher.

We then use a ranking SVM [51, 107] to train prediction models by combining content and network features (rankings). **META1** combines the following rankings: URL-RANK, HTG-RANK and PROLIFIC. **META2** combines the following rankings: URL-RANK, HTG-RANK, WT-COM-BON and MIX-PATH.

5.3.6 Summary of the methods

Table 5.1 is the list of our proposed methods and some baseline methods. HTG-RANK and URL-RANK are unsupervised and use content features. PROLIFIC and BON use simple network features from a single network; BON utilizes the Bonacich path metric. META1 is a supervised method that combines content and simple network features. The main contribution of our research are two unsupervised methods WT-COM-BON and MIX-PATH, all of which are based on *hybrid network features*, i.e., they combine features of the three networks. The baseline method UNW-COM utilized the hybrid network in an unweighted way. Finally META2 is a supervised method that combines the hybrid network features with content features. We will evaluate different methods in subsequent sections.

Table 5.1: Different types of methods

Content Feature			HTG-RANK
			URL-RANK
Network Feature	Single Network	Naive	PROLIFIC
		Path-based	BON
	Hybrid Network	Unweighted	UNW-COM
		Weighted	WT-COM-BON
			MIX-PATH
Meta	HTG-RANK + URL-RANK + PROLIFIC		META1
	HTG-RANK + URL-RANK + WT-COM-BON + MIX-PATH		META2

5.4 Evaluation Dataset and Metrics

5.4.1 Data Collection

There have been several successful efforts to construct a proxy graph that characterizes the structure of a real network [36, 61]. For this experiment, our objective was different. It was to construct a dataset that reflected a comprehensive history of user interaction and tweet content, over an extended period, for a significant number of active users, given the strict limitations imposed by the Twitter API. We constructed a network of 15,000 users, as well as all their follower (friend) associations within this subnetwork. In choosing these 15,000 users, we focused on active users. Our premise is that the active users generate the most content and have the greatest influence. Thus, following the largest number possible (15,000) of active users provided us with a dataset that captured a majority of the activity that would have had an influence on these 15,000 users. We note that had we constructed a 15,000 user dataset to reflect the typical distribution of users in the network, we may have been severely limited in our ability to capture a majority of the relevant activity.

We used the Twitter API to construct the network in the following way: Starting from a seed *active* user, we expanded her follower network and added further active users until we reached 15,000 active users. The test for an active user was as follows based on their most recent 100 tweets: (1) The user should have an average minimum tweet frequency of one tweet per day in this time period. (2)There was at least one retweet in the most recent 100 tweets. We used the twitter streaming

API to collect all tweets published by the 15K active users between April 25, 2011 and June 25, 2011.

Retweeting is identified by the use of *RT @username* in tweets. Mentioning is identified by *@username* in the tweet content, after excluding *RT @username* . We built up the retweet and mention network by extracting users being retweeted or mentioned in each tweet. Hashtags identified by *#hashtag* and URLs were also extracted. Since username and hashtag are case insensitive, we transformed all usernames and hashtags to lowercase.

Test Dataset and Training Dataset:

We used the first month data (from April 25th to May 25th) as the training dataset and we obtained the ground truth from the second month data (from May 26 to June 25) and used it as the test dataset. We picked the sets of microblog users who had ground truth in the test dataset for evaluation. 4257 users had retweet ground truth and 7296 users had mention ground truth. The average number of ground truth (retweeters) for the 4257 users is 4.56, and the average number of ground truth (mentioners) for the 7296 users is 8.12.

For supervised learning, we selected those microblog users that contain ground truth from May 15 to May 25 as focal training users. The features of the users that were used to produce the training pairs for the ranking SVM for each focal training user were collected in the preceding time interval from April 25 to May 15. There was no overlap between the training and testing time interval. Similarly there was no overlap in the time interval for feature collection and to obtain the training ground truth.

Table 5.2: Statistics of the twitter experiment data set

Time range	04/25/11–06/25/11
Number of active users	15,000
Number of tweets for the active users	10,979,278
Number of edges of following relationship within the 15K active users	3,293,840
Number of retweets by and of the 15K active users, excluding to-self retweets	147,970
Number of mentions by and of the 15K active users, excluding to-self mentions	584,597
Number of appearances of hashtags	3,616,614
Number of distinct hashtags	302,628
Number of appearances of URLs	3,622,992
Number of distinct URLs	2,611,550

5.4.2 Metrics and Parameters

Mean Average Precision (MAP) is widely used for evaluating ranking methods; it provides a single-figure measure of quality across recall levels. MAP has been shown to have especially good discrimination and stability [69] and we report on the values of MAP. We also use Normalized Discounted Cumulative Gain (NDCG) [44] for evaluation. NDCG is also a measure commonly used for evaluating the results of ranking methods. The NDCG value of a ranking list at position i is calculated as:

$$NDCG@i = Z_i \sum_{j=1}^{i} \frac{2^{r(j)} - 1}{log(1 + j)}$$

where $r(j)$ is the rating for the jth item and Z_i is a normalization constant. Z_i is chosen so that the NDCG score for a perfect ranking is 1. In our experiments we measured NDCG at the positions of 5 and 10.

In the following evaluations, when we do not specify the values, the default chosen values of the parameters are as follows: $K = 20$; $\alpha = 0.00005$; $\beta = 1.0$; $\varepsilon = 3.5$; for *MIX-PATH*, the path length factor $\ell(\cdot) = 1.0$ when path length is 1 and $\ell(\cdot) = 0.01$ when path length is 2.

5.4.3 Network Properties

Figure 5.4 reports on the average numbers of followers, retweeters, mentioners for each twitterer, and the average numbers of their overlaps. The figure shows that the average number of followers for each twitterer is 219.59, the average number of retweeters is 3.71, and the average number of mentioners is 8.80. And it also shows the intersections of the retweeters, mentioners and followers. From the intersections, we can calculate that for a user, 1.3% of her followers retweet her tweets in the 2-month dataset, 2.4% of her followers mention her in the 2-month dataset, and 0.7% of her followers both retweet her tweets and mention her in the 2-month dataset.

Figure 5.5 reports on the follower network degree distribution within the 15K active users. Figure 5.5(a) is on log-linear scale, while Figure 5.5(b) is on log-log

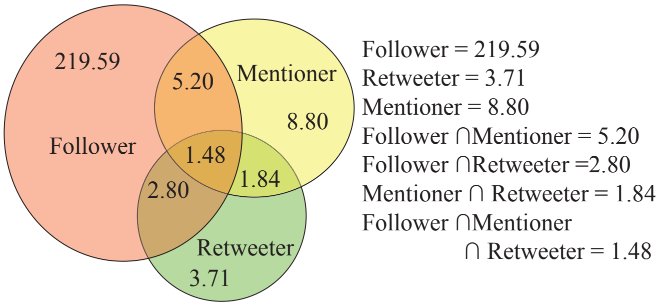
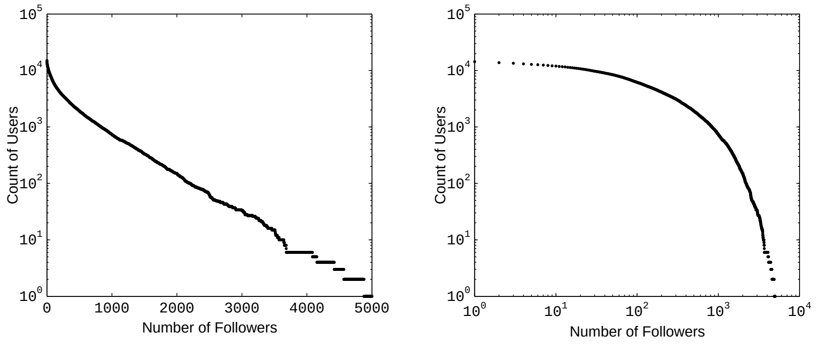


Figure 5.4: Average numbers of followers, retweeters, mentioners from the 15K users and the 2-month dataset for each twitterer (excluding to-self).

scale. The distribution has an approximately straight-line form on the log-linear scale for majority part of the data, while it only shows part of straight-line form on the log-log scale (in the beginning and the tail), which means that the follower network within the 15K active users is more close to an exponential degree distribution in general, but for degree less than 100 or greater than 3000 it is more close to a power-law degree distribution. Figure 5.6 reports on the retweet network degree distribution and Figure 5.7 reports on the mention network degree distribution within the 15K active users in the 2-month period. Both retweet network and mention network have a power-law degree distribution (note the log-log scales used in Figure 5.6 and Figure 5.7). The social network usually has the power-law characteristic. The way that we used for sampling the data from Twitter might have some influence on the network degree distributions, but it would not have significant effect on our evaluation results.



(a) log-linear scale

(b) log-log scale

Figure 5.5: Follower network degree distribution within the 15K active users. X axis is the number of followers; Y axis is the number of users that have the number of followers greater than or equal to the corresponding value on X axis.

5.4.4 Hybrid Network Correlation

We know that there are some correlation between different types of network. As discussed in section 5.3, we used the metric of spearman correlation to evaluate their correlation. For two network adjacency matrices A and B, we calculated the spearman correlations of the corresponding rows of the two matrices, and then used the average value to represent the correlation of the two matrices.

Table 5.3 reports on the spearman correlation of different types of networks to the retweet network constructed from the first 30-day training data. Table 5.4 reports on the spearman correlation of different types of networks to the mention network constructed from the first 30-day training data. From the two tables we have the following observations and indications:

- The correlation between R and M is stronger than the correlation between R and F^*, and stronger than correlation between M and F^*. This indicates that

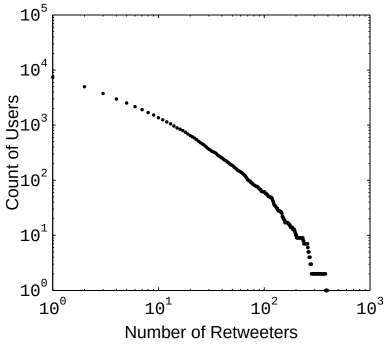


Figure 5.6: Retweet network degree distribution within the 15K active users. X axis is the number of Retweeters; Y axis is the number of users that have the number of Retweeters greater than or equal to the corresponding value on X axis.

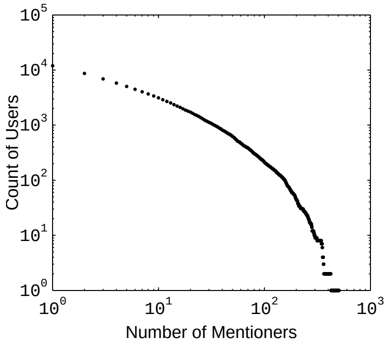


Figure 5.7: Mention network degree distribution within the 15K active users. X axis is the number of mentioners; Y axis is the number of users that have the number of mentioners greater than or equal to the corresponding value on X axis.

to predict retweeters, the feature of mentioners are more important than the feature of followers; similarly to predict mentioners, the feature of retweeters is more important than the feature of followers.

- The correlation between R and RF^* is stronger than the correlation between R and F^*R. This indicates that for a user, the followers of his retweeters are more likely to retweet him than the retweeters of his followers.

- The correlation between R and MF^* is stronger than the correlation between R and F^*M. This indicates that for a user, the followers of his mentioners are more likely to retweet him than the mentioners of his followers.

Table 5.3: Correlation to retweet network adjacency matrix

	R	M	F^*	RR	MM	F^*F^*	RM	MR	RF^*	F^*R	MF^*	F^*M
R	1.000	0.653	0.222	0.638	0.495	0.073	0.621	0.526	0.601	0.130	0.443	0.116

Table 5.4: Correlation to mention network adjacency matrix

	R	M	F^*	RR	MM	F^*F^*	RM	MR	RF^*	F^*R	MF^*	F^*M
M	0.653	1.000	0.222	0.578	0.475	0.076	0.571	0.489	0.535	0.128	0.444	0.120

- The correlation between M and RF^* is stronger than the correlation between M and F^*R. This indicates that for a user, the followers of his retweeters are more likely to mention him than the retweeters of his followers.

- The correlation between M and MF^* is stronger than the correlation between M and F^*M. This indicates that for a user, the followers of his mentioners are more likely to mention him than the mentioners of his followers.

5.5 Evaluation Results

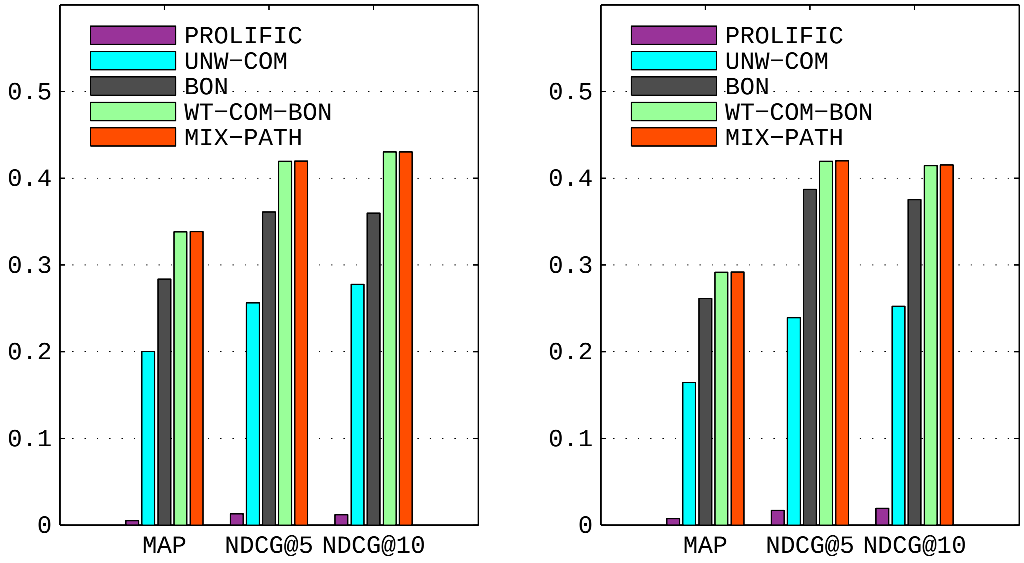
We first report results on the prediction accuracy of the unsupervised methods on the entire ground truth. We then compare with the supervised methods. Finally, we consider the more challenging subset of novel retweeters and mentioners, i.e., they have not retweeted or mentioned the focal user in history, and retweeters who are not followers.

5.5.1 Results on the Entire Ground Truth

Figure 5.8(a) reports on the MAP, NDCG@5 and NDCG@10 for the 5 unsupervised methods for retweet prediction. Figure 5.8(b) reports on the results for mention prediction. Both 5.8(a) and 5.8(b) demonstrate that *WT-COM-BON* and *MIX-PATH* have good prediction accuracy and dominate the three baseline methods *PROLIFIC*, *BON*, and *UNW-COM*.

The prediction accuracy for *PROLIFIC* is very low. This suggests that simply picking those who retweet mention frequently is not a good strategy. *BON* exploits a simple network while *UNW-COM* creates a hybrid network using a naive unweighted union. *BON* dominating *UNW-COM* suggests that a naive combination does not improve and can actually degrade performance compared to exploiting a simple network.

The performance accuracy appears to be higher for retweet prediction compared to mention prediction. We note that retweets reflect the influence of both a topic and a focal user, and this may explain the improved prediction accuracy.



(a) Retweet ground truth

(b) Mention ground truth

Figure 5.8: The performance of the prediction methods on all of the ground truth.

5.5.2 Incorporating Content Features by Supervised Learning

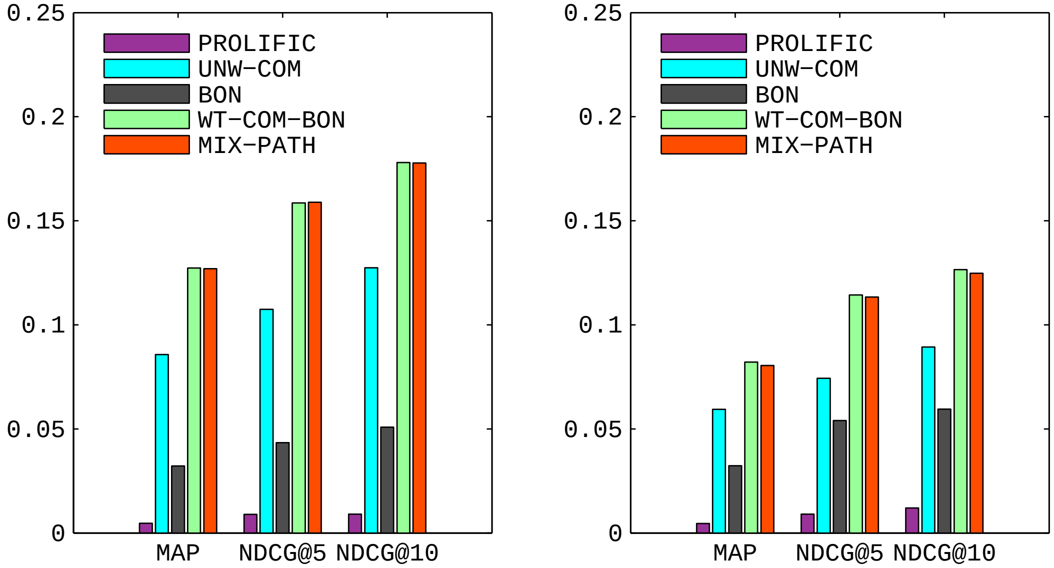
Table 5.5 reports on the supervised learning methods *META1* and *META2* that incorporate content features over the entire ground truth for retweet prediction. As expected, *META1* outperforms *URL-RANK*, *HTG-RANK* and *PROLIFIC*. We note with interest that *META2* does not appear to improve on *WT-COM-BON* or *MIX-PATH*; we report on the fourth decimal place to reflect that the results are not identical. This suggests that the hybrid network actually capture effective content features based on shared hashtags and URLs; thus, **META2** is not able to exploit additional content features.

Table 5.5: The performance of the supervised learning methods with comparison to other methods on all of the ground truth for retweet prediction

	MAP	NDCG5	NDCG10
PROLIFIC	0.0051	0.0130	0.0120
URL-RANK	0.1631	0.2194	0.2225
HTG-RANK	0.1427	0.1888	0.1991
META1	0.1720	0.2289	0.2344
META2	0.3331	0.4140	0.4220
WT-COM-BON	0.3382	0.4196	0.4303
MIX-PATH	0.3384	0.4198	0.4302

5.5.3 Results on the Ground Truth from Novel Retweeters and Novel Mentioners

For a focal user, those novel retweeters who will retweet his tweets in the future but did not retweet his tweets in the history would be more difficult to identify than normal retweeters; similarly, those novel mentioners who will mention him in the future but did not mention him in the history would also be more difficult to identify than normal mentioners. We want to evaluate on the prediction accuracy for novel retweeters and novel mentioners. For each method, when we target the ground truth on novel retweeters for a focal user, those users who retweeted the focal user's tweets in the history will be excluded from the prediction; similarly, when we target the



(a) Novel retweeters ground truth

(b) Novel mentioners ground truth

Figure 5.9: The performance of the methods on the ground truth from novel retweeters and novel mentioners.

ground truth on the novel mentioners for a focal user, those users who mentioned the focal user in the history will be excluded from the prediction.

Figure 5.9(a) reports on the MAP, NDCG@5 and NDCG@10 for the methods for retweet prediction on novel retweeters ground truth. Figure 5.9(b) reports on the results for mention prediction on novel mentioners ground truth. As expected, the prediction accuracy on novel ground truth is much lower than on normal ground truth. However, for the novel ground truth, our proposed methods perform much better than the methods that do not use the hybrid networks. For novel retweeters ground truth, *BON* has an MAP value of 0.032, while our proposed methods *WT-COM-BON* and *MIX-PATH* both have an MAP value of 0.127, which is almost four times as big as that of *BON*; *BON* has an NDCG@5 value of 0.043, while our proposed methods *WT-COM-BON* and *MIX-PATH* both have an MAP value of 0.159, which is more than three times as big as that of *BON*. For novel mentioners ground truth, *BON* has an MAP value of 0.032, while our proposed methods *WT-COM-BON* and *MIX-PATH* have an MAP value of 0.082 and 0.081 respectively, which is more than twice as big as that of *BON*; *BON* has an NDCG@5 value of 0.054, while our proposed methods *WT-COM-BON* and *MIX-PATH* have an MAP value of 0.114 and 0.113 respectively, which is also more than twice as big as that of *BON*. Both figures also show that *UNW-COM* dominate *BON* for novel ground truth, which is opposite for normal ground truth. It tells us that utilizing hybrid networks is especially important for the prediction of future novel linkers in the environment where hybrid networks exist.

Tables 5.6 reports on the results of the supervised learning methods *META1* and *META2* with comparison to the methods that are combined to the meta methods, on the ground truth from novel retweeters. Like results on all of the ground truth, the supervised learning methods do not show to improve *WT-COM-BON* and *MIX-PATH* on novel ground truth.

Table 5.6: The performance of the supervised learning method with comparison to other methods on the ground truth from novel retweeters.

	MAP	NDCG5	NDCG10
PROLIFIC	0.0047	0.0090	0.0090
URL-RANK	0.0404	0.0536	0.0591
HTG-RANK	0.0589	0.0761	0.0870
META1	0.0466	0.0597	0.0675
META2	0.1123	0.1401	0.1561
WT-COM-BON	0.1273	0.1586	0.1780
MIX-PATH	0.1270	0.1590	0.1778

Chapter 6

Case Studies

In this chapter, we present two recommendation case studies based on the prediction work in the previous chapters. The results are presented in [108, 109].

6.1 Recommendation in Blogosphere

As one case, we present a recommendation system for social media that draws upon monitoring and prediction methods. We use historical posts on some focal topic or historical links to a focal blog channel to recommend a set of authors to follow. Such a system would be useful for brand managers interested in monitoring conversations about their products. Our recommendations are based on a prediction system that trains a ranking Support Vector Machine (RSVM) using multiple features including the content of a post, similarity between posts, links between posts and/or blog channels, and links to external websites. We solve two problems, Future Author Prediction (FAP) in Chapter 3 and Future Link Prediction (FLP) in Chapter 4, and apply the prediction outcome to make recommendations. Using an extensive experimental evaluation on a blog dataset, we demonstrate the quality and value of our recommendations.

6.1.1 Introduction

During a critical period such as a product release, it would be useful if a brand manager could be provided with recommendations about who will participate in social media conversations. This information could be used to actively participate in those conversations, and potentially to contact authors proactively, to provide them with accurate information or to address any concerns. We illustrate the potential benefits of such a recommendation using an example set of posts related to the release of the Blackberry Storm. Figure 6.1 is a Google Insights graph showing the relative search volume for the phrase *Blackberry Storm*, which was launched in October 2008. The search volume gradually increases in September and there is a peak (`B event`) in October and another peak (`A event`) in November. The Google Insights graph clearly indicates that there is a growing interest in this particular product.

Figure 6.2 reports on the distribution of posts about *Blackberry Storm* that occurred in a two-month blog dataset from August to September 2008 that we use for our experiments. To identify the posts relevant to this topic, we used document similarity and event detection [89]. As can be seen, there were more posts discussing the focal topic of *Blackberry Storm* in September than in August, and the Google Insights data indicates a corresponding growth. Many of the posts about *Blackberry Storm* discuss features of the new product, and compare these features to other products. Several of these posts contain very similar descriptions indicating that information is flowing from a uniform source and then diffusing through social media.

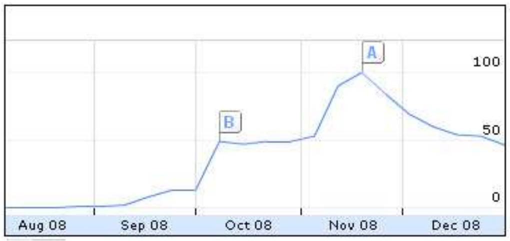


Figure 6.1: Google Insights for “Blackberry Storm” in 2008, where *“A” event is “Review: BlackBerry Storm” and “B” event is “BlackBerry Storm Launches; BlackBerry Bold Hides”.*

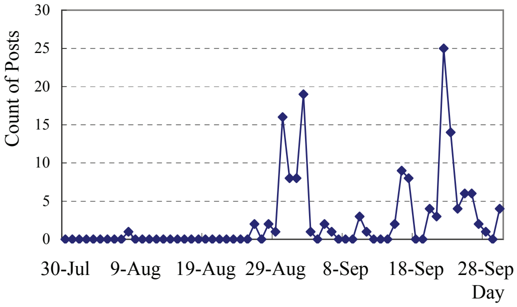


Figure 6.2: Distribution of the number of blog posts talking about “Blackberry Storm” in our dataset.

For brand monitoring, it is valuable to know, in advance, not only how many people will be involved, but who will be involved in a focal topic. From Figure 6.2 we can see that in August, *Blackberry Storm* is an emerging topic. This implies that the authors who post are *Novel Authors*, i.e., they had not posted on this topic before. As the topic diffuses, new posts may continue to appear from additional *Novel Authors* or there may be repeat posts from authors.

In addition to post content, it is also important to examine links to a focal blog channel, e.g., links to the authors on the focal topic. These links may attract new readers. The most useful recommendations are links that originate from *Novel Linkers* who have not historically linked to the focal blog channel.

Recommending *Novel Authors* and *Novel Linkers* before they post would be useful for brand monitoring. If a manager knows who will create future relevant posts or links, then actions can be taken, such as trying to mitigate negative opinion in the spread of information by engaging in a conversation.

A brand manager would also benefit from being able to identify authors who will post on diverse subtopics within the focal topic or subject area is particularly useful since these authors provide fresh, potentially diverse information about the perception of the product. If a brand manager listens to the same authors all of the time, then the manager may not truly understand how the crowd as a whole is responding to the product. As an example, interesting posts about the *Blackberry Storm* were generated by: (1) authors who were employees of the company; (2) a blog pushing technology related information to teens; (3) a marketing and advertising site; (4) a portal providing coupons for mothers; etc. Recommending such diverse authors who have different views on the focal topic to brand managers would help them to build a larger picture of how their brand is being perceived.

6.1.2 Methodology for Social Media Recommendation

Figure 6.3 illustrates the architecture and methodology for recommendation. A historical set of posts prior to time T_q are used to construct a *Profile* for the author of each blog channel, i.e., stream of posts. The profile represents the cumulative

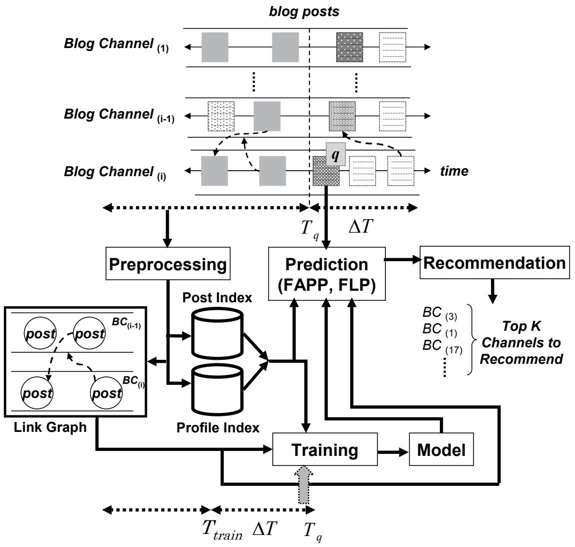


Figure 6.3: System architecture for recommendation.

content of all posts; a temporal decay model is used to update the profile. Further, a *Link Graph* of links between pairs of posts or links from a post to a blog channel, as well as links to pages outside the dataset is created.

The next stage involves two prediction problems: (1) Given a focal query post on some topic T, Future Author Prediction (FAP) [107] will predict an author whose blog channel will contain a future post that is relevant to T, and (2) Given a focal query channel, the Future Link Prediction (FLP) [110] will predict an author whose blog channel will contain a future post with a link to the focal channel. The detailed definitions of the two problems can be found in Section 3.2 and Section 4.2 respectively. Our final step is to make a recommendation; we consider the following types of recommendations:

- *Novel Author* or *Novel Linker*: Among the predictions for FAP and FLP, an author who has not posted on the focal topic in the historical data, or an author who has not linked to a focal blog channel in the past is considered a high value recommendation. This is because it would be unlikely that someone with an interest in the focal topic or channel would be able to easily identify these authors or linkers.

- *Diverse Profile*: Consider an author whose profile is dissimilar to the profiles of current authors on a focal topic. If this author were to post on the focal topic or link to a focal blog channel with posts on the focal topic, then this author may have greater impact since their profile diversity may represent a different expertise or a different following. As before, such an author or linker with a diverse profile is harder to identify if we only use profile similarity.
- *Combined Recommendation*: Consider a focal blog channel and a query post on topic T that occurs on the focal channel. From among the recommendations, an author or blog channel who will post on the focal topic T and will link to a focal blog channel will be a high value prediction. Such a combined prediction (recommendation) would be valuable because it implies both an interest in the focal topic T accompanied by an interest in a particular focal blog channel.

6.1.2.1 Solution Approaches

We have developed multiple solutions for both FAP and FLP problems. We present the methods for FAP problem in Section 3.3 and for FLP problem in Section 4.3. For both problems, we use the ranking SVM [51] to train our prediction model. How we train the ranking SVM is described in Section 3.3.3. For FAP and FLP, the features for training the ranking SVM are different, which are described in Section 3.3.3 and Secition 4.3.4 respectively. We use the prediction output from the ranking SVM For both FAP and FLP for recommendations in this case study.

6.1.3 Experimental Evaluation

We evaluate our recommendation system on a dataset provided by Spinn3r.com. The property of the original dataset and how we preprocess the dataset as well as the statistics of the experimental dataset after preprocessing are described in Section 3.4.1.1.

In Section 3.4 and Section 4.4, we have evaluated different prediction methods on multiple time periods of test datasets. In this section, for evaluating the recommendations, we focus on one time period of test datasets. We select 10 days from September 1 to 10 to be the period of test datasets. Our training dataset was 31 days from July 30 to August 31, which is the same as those in Section 3.4 and Section 4.4.

For the *FAP* task, we selected 861 query posts on September 1. The posts have at least one ground truth blog channel in the test dataset. To identify these ground truth blog channels, we used the Okapi BM25 weighting function [85] to calculate document similarity between the query post and all posts in the test dataset. We set a similarity score threshold of 130 to determine the ground truth, i.e., any blog channel that has a post in the test dataset that meets or exceeds the similarity threshold is a ground truth blog channel. More details are described in Section 3.4.

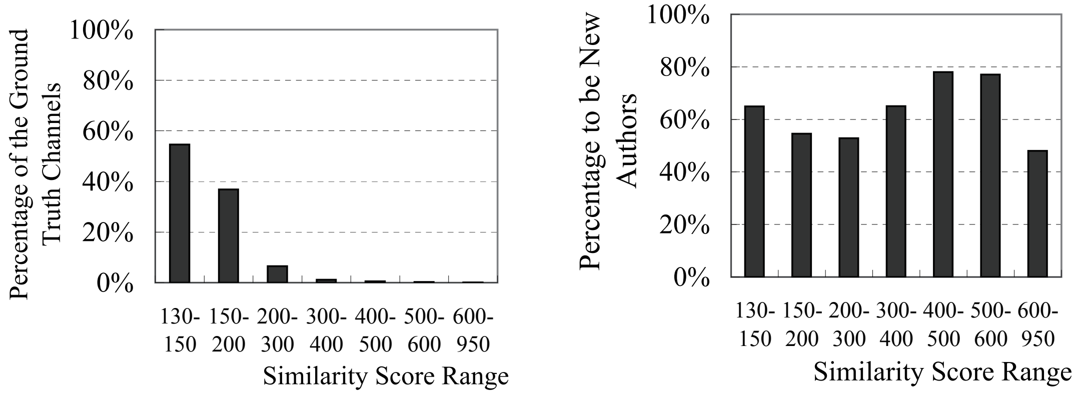
For the *FLP* task, we selected 3636 focal blog channels that contain at least 1 ground truth blog channel in the 10 day test dataset, i.e. these blog channels have at least 1 in-link in the test dataset. More details are described in Section 4.4.

6.1.3.1 Novel Author Recommendation

Recall that we set a similarity score threshold of BM Okapi 130 to determine the ground truth posts in the test dataset. We use the same threshold to identify a *Novel Author*, i.e., a *Novel Author* is one whose historical posts *do not include* any posts that meet or exceed the similarity score threshold with respect to the query post.

Figure 6.4(a) is the distribution of ground truth blog channels in the test dataset. There are 40328 ground truth blog channels; they are binned based on their similarity score with respect to the query post. We observe that as the similarity score increases the percentage of the ground truth decreases. Of the 40328 ground truth blog channels, approximately 60% are *Novel Authors.* Figure 6.4(b) illustrates the distribution of *Novel Authors* also binned by their similarity score to the query post. We note with interest that posts by *Novel Authors* are more evenly spread among the bins.

Next, we consider the precision and recall of the true positive predictions for FAP which are the basis of our recommendation. When K = 20, 9.0% of the recommendations are *Novel Authors.* Figure 6.5(a) and 6.5(b) illustrate recall of the FAP true positive predictions for K=1000. Figure 6.5(a) illustrates the recall of the



(a) Ground Truth Distribution

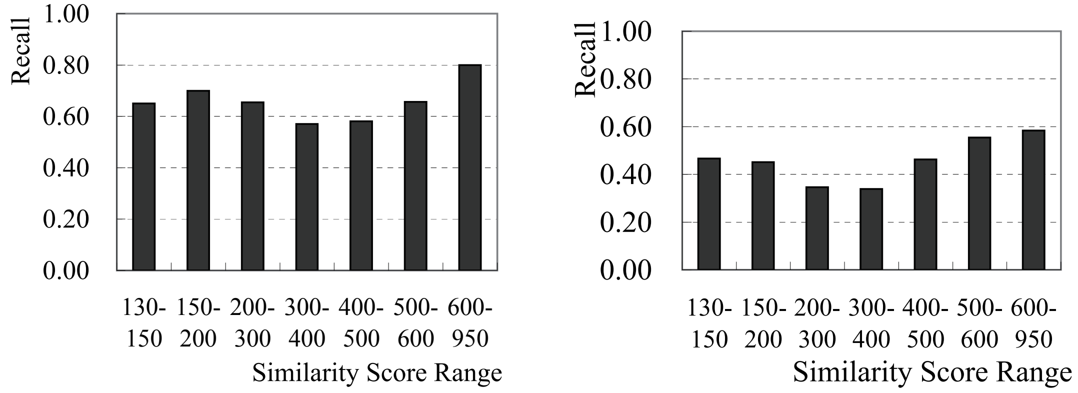
(b) Novel Author Distribution

Figure 6.4: Distribution of ground truth blog channels and distribution of Novel Authors with different similarity scores between the ground truth posts and the query posts.

true positives with respect to the entire ground truth and Figure 6.5(b) illustrates the recall of the true positives with respect to the *Novel Authors* in the ground truth. As expected, the recall for the entire ground truth is higher than the recall for *Novel Authors*. We note with interest that for *Novel Authors*, while the recall is highest for the bins with the highest similarity scores, the recall is still high even for bins with less similarity scores. This implies that our system can make a diverse range of recommendations.

6.1.3.2 Novel Linker

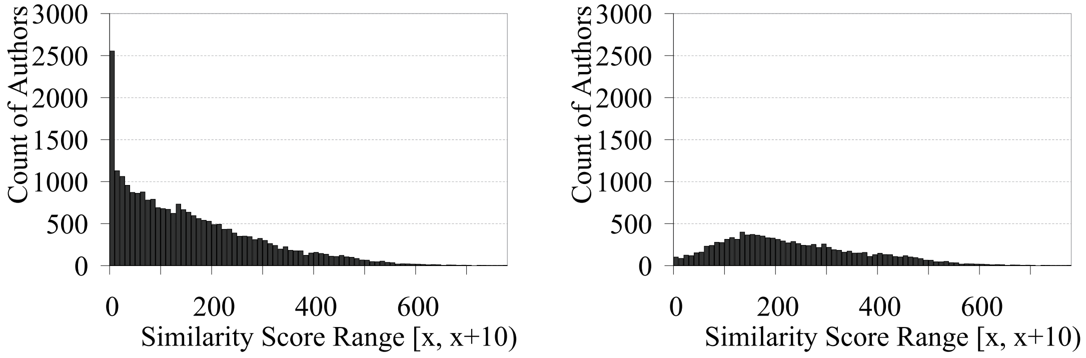
For the *FLP* recommendation, given a focal blog channel, we want to recommend other blog channels that link to the focal blog channel. Clearly, channels that have historical links are easier to predict. However, 76.1% of the ground truth blog



(a) The entire ground truth

(b) Novel Authors

Figure 6.5: Recall for the *FAP* task.



(a) All the Novel Authors

(b) True Positive Novel Authors

Figure 6.6: Distribution of the similarity scores of the query posts to profiles of all the Novel Authors and to the profiles of the true positive Novel Authors.

channels for *FLP* are *Novel Linkers*. If we consider the Top K true positive predictions for the FLP task, when K = 20, 30.4% of the true positives are *Novel Linkers* that do not have historical links to the focal blog channel. Further, 14.6% of the true positives do not have historical direct links or paths to the focal blog channels. The recall value for K=20 is 0.452. Our ability to recommend almost 50% of the *Novel Linkers* illustrates the accuracy of the FLP prediction. It also illustrates the benefit of our recommendation approach since *Novel Linkers* are difficult to identify.

6.1.3.3 Diverse Profiles

Diversity of the profiles (historical posts) of authors is a key factor in a successful high value recommendation. Figure 6.6(a) reports on the distribution of the similarity scores between the profile of each Novel Author when compared to the query post. We observe with interest that many profiles have low similarity scores, thus making prediction more difficult. Figure 6.6(b) reports on the distribution of the similarity scores between a profile of a true positive predicted Novel Author with respect to the query post. We note with interest that the recommendations are evenly spread across the range of similarity scores. We further note that the peak of the recommendations are at lower scores. This further illustrates that we are able to make diverse recommendations based on author profiles.

6.1.3.4 Combined Predictions

We also consider a *Combined* author that will post on the focal topic T and will link to a focal blog channel. This is a high value prediction since it implies both an interest in the topic T and in a particular focal blog channel. Note that this is the intersection of the predictions of FAP and FLP. For the 861 query posts, there are 244 authors who both post on a topic and link to the corresponding focal blog channel. For K=100 recommendations, FAP recommended 97 *Combined* authors and FLP recommended 81. Combining FAP and FLP can therefore lead to excellent recommendations.

6.2 Recommendation in Microblogs

As another case, we present a microblog recommendation system that can help monitor users, track conversations, and potentially improve diffusion impact. Given a Twitter network of active users and their followers, and historical activity of tweets, retweets and mentions, we build upon a prediction tool to predict the Top K users who will retweet or mention a focal user, in the future. The retweet and mention predictions are presented in Chapter 5. We develop personalized recommendations for each focal user. We identify characteristics of focal users such as the size of the follower network, or the level of sentiment averaged over all tweets; both have an impact on the quality of personalized recommendations. We use (high) betweenness centrality as a proxy of attractive users to target when making recommendations. Our recommendations successfully identify a greater fraction of users with higher betweenness centrality, in comparison to the overall distribution of betweenness centrality of the ground truth users for some focal user.

6.2.1 Introduction

The usage of social media has grown considerably in recent years, with microblogging sites being an important area of growth. On a site such as Twitter, one can follow a user and read their tweets. One can initiate a new conversation by tweeting or one can interact by mentioning a user. One can also participate in the diffusion of a topic by retweeting. Influence in a microblog can be captured in multiple ways. One can generate a lot of content or befriend a lot of users but this may not lead to a large follower network or increase diffusion. Someone who has high betweenness centrality, whose tweets diffuse rapidly or widely outside her immediate follower network, or someone who is mentioned frequently by other users may, is typically considered influential. Other factors such as the level of sentiment or persuasiveness may also play a role in diffusion.

We are interested in analyzing both diffusion and influence in microblogs such as Twitter, from the individual or personalized perspective. We want to understand who will be influenced by a particular focal user. Given a Twitter network of active users and their followers, and their historical activity of tweets, retweets and mentions, we build upon a prediction tool that uses history to predict the Top K users who will retweet or mention a focal user, in the future (see Chapter 5). Our objective is to make high quality personalized recommendations for each focal user.

Social network and social influence analysis has drawn a lot of research interest. Previous research on social influence had a focus on the measurement of social influence [21] or attempted to maximize user influence [22, 53] at the aggregate level. Our objective is to track those users who will likely be influenced by an individual focal user and to improve the impact of the focal user.

We identify characteristics of focal users such as the size of the follower network and the level of sentiment averaged over all tweets. We demonstrate that these features have an impact on the quality of personalized recommendations, i.e., accuracy of predictions. As the focal user's follower network increases, prediction accuracy decreases. In contrast, we can improve prediction quality for focal users with higher levels of positive sentiment. We note that the focal users with higher levels of positive sentiment appear to have a larger following. Despite a larger following having been shown to decrease prediction accuracy, we are nevertheless able to successfully recommend users who will retweet the more positive focal user (in the future ground truth) with greater accuracy.

We use (high) betweenness centrality as a proxy of attractive and potentially influential users to target when making recommendations. Our recommendations successfully identify a greater fraction of users with higher betweenness centrality, in comparison to the overall distribution of centrality among the ground truth users.

In summary, despite the difficulty of diffusion and influence prediction in evolving and noisy microblog networks, we have been successful in making personalized recommendations with improved accuracy for focal users with high(er) positive sentiment levels. We also are able to successfully recommend users with potentially greater influence (high betweenness centrality).

6.2.2 Solution Approach

In Chapter 5, we have presented a prediction model for retweet prediction and mention prediction, and have proposed two approximate approaches, which are demonstrated to perform better than other alternative methods. In this recommendation study, we choose one of the best approaches, **WT-COM-BON** for prediction, and use the output for recommendation.

6.2.3 Experimental Evaluation

6.2.3.1 Dataset and Metrics

In Section 5.4.1, we have described how we collected the 15000 active users and the follower network, and two months of their tweets, and how we preprocessed the tweets to get the information. For this section, rather than using all of the 15000 users for evaluation, we did some more filtering. We know that as we only collected a subset of the users, some of the users may only have a small fraction of her friends or followers in the subset of the users. We used a threshold $X\%$ to filter out those users in the subset by the following way:

- First get a set of the users who has at least $X\%$ of friends and also at least $X\%$ followers from the 15K users. Label this set of users as S.
- Repeat the following loop until the number of users in S is stable, i.e., $|S|$ does not change:

  For each user in S, if the number of her friends or the number of her followers from S is less than $X\%$ of the total number of his friends or the total number of his followers, remove this user from S.
- Return the set of users S.

We set the threshold $X\% = 2.4\%$. Our crawling statistics shows that 40% of Twitter users were "active users", and we only collected "active users". So with this threshold, we got a subset of users with at least around 6% of their active followers and 6% of their active friends in the subset. We used the first month of our data (

from April 25th to May 25th) as a training dataset and we used the second month data (from May 26 to June 25) as a test dataset where we obtained the ground truth. We picked the sets of microblog users who had ground truth in the test dataset for evaluation. 2728 users had retweet ground truth and 4571 users had mention ground truth. The average number of ground truth (retweeters) for the 2728 users is 4.23, and the average number of ground truth (mentioners) for the 4571 users is 8.64.

The metric that we used for evaluation is MAP (Mean Average Precision). MAP is widely used for evaluating for ranking methods. We set the K value to be 20.

6.2.3.2 Impact of User Network

Networking features such as the count of friends and followers, both from the global counts registered on the Twitter profile, as well as the local counts computed in our dataset, were found to be highly significant when creating a model to explain variants of user behavior and the impact of diffusion effectiveness, as reported in [96]. The same holds true for the accuracy of future retweet and mention prediction.

Figure 6.7 reports on the prediction accuracy for the focal users whose total number of followers is less than, or is greater than, the average number of followers of the focal users. The left part of the figure is for retweet prediction and the right is for mention prediction. The figure demonstrates that it is more difficult to predict for focal users with a larger following. When a user has more followers, more people will potentially read their tweets and retweet or mention her in the future. Some of

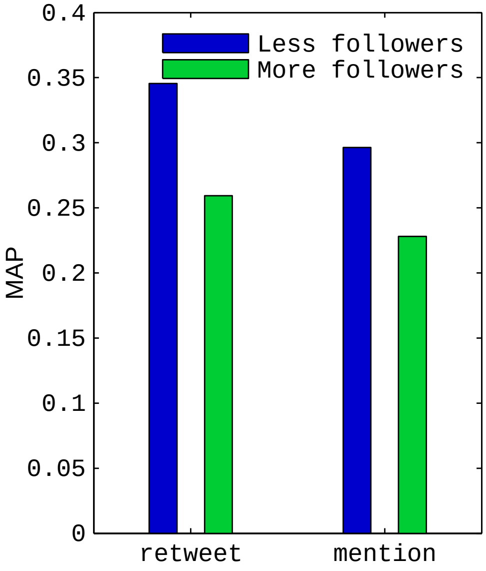


Figure 6.7: Prediction accuracy for the focal users with the number of followers less than and greater than the average number of followers of all focal users. The left part of the figure is for the focal users of retweet prediction; the right part is for the focal users of mention prediction.

the future users will be novel users who did not retweet her in the past. Both cases increase the difficulty of prediction.

6.2.3.3 Impact of Sentiment

Sentiment has also been widely identified as an important factor of influence and diffusion. We used a dataset and tool [71] trained for sentiment detection in tweets. In the training dataset, tweets containing positive emoticons like ":)" but not negative emoticons were labeled as positive, and tweets containing negative emoticons like ":(" but not positive emoticons were labeled as negative. A Na ive Bayes classifier (NBC) was constructed using the sentiment training dataset of 232K negative tweets and 232K positive tweets. We then used the NBC to classify our training dataset to assign a sentiment score to each tweet, in the range of [-1, +1]. Finally, we averaged the sentiment score over all the tweets of a user to determine a level of sentiment. Figure 6.8 reports on the distribution of the sentiment scores for each of the tweets of our dataset. Figure 6.9 reports on the distribution of the user sentiment level computed over all the tweets of each user.

Figure 6.10 reports on the comparison of focal users with a sentiment level less than, and greater than, the average sentiment level of the focal users, for retweet prediction. Figure 6.10(a) compares the prediction accuracy while Figure 6.10(b) presents the number of followers. Figure 6.10(b) shows that users with a more positive sentiment level are more likely to attract a larger follower network. We have shown in a previous result, that it is more difficult to predict for focal users with more followers. However, for retweet prediction, Figure 6.10(a) shows that we can predict future users for focal users with more positive sentiment, with higher prediction accuracy. For example, for very positive focal users with user sentiment level > 0.9, the MAP value for retweet prediction is 0.395. In contrast, for very negative focal users with user sentiment score < 0.2, the MAP value for retweet prediction has reduced drastically is 0.253.

6.2.3.4 Impact of Centrality

The betweenness centrality of a node v in a network is defined by the expression:

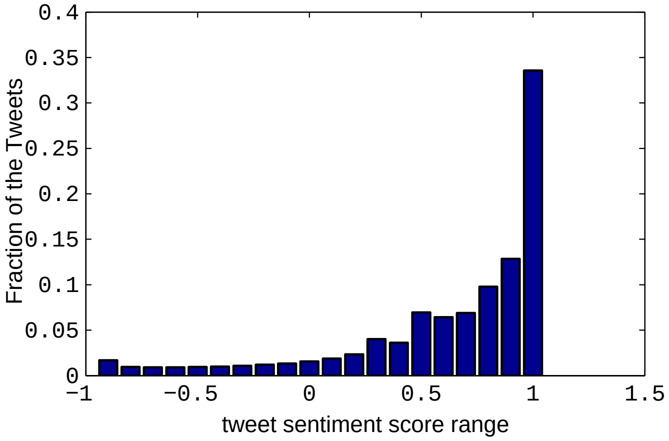


Figure 6.8: The distribution of the sentiment scores for all of the tweets.

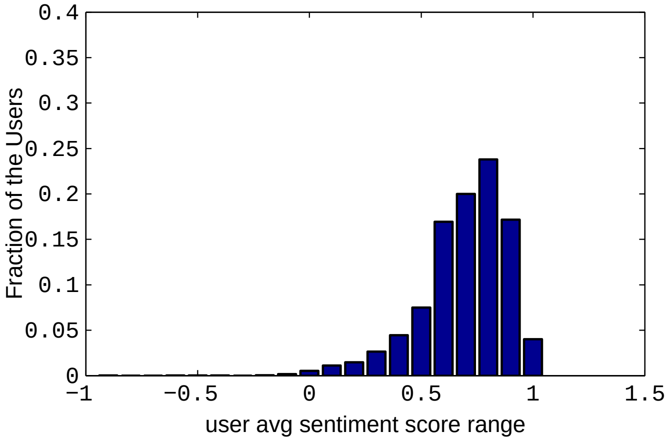
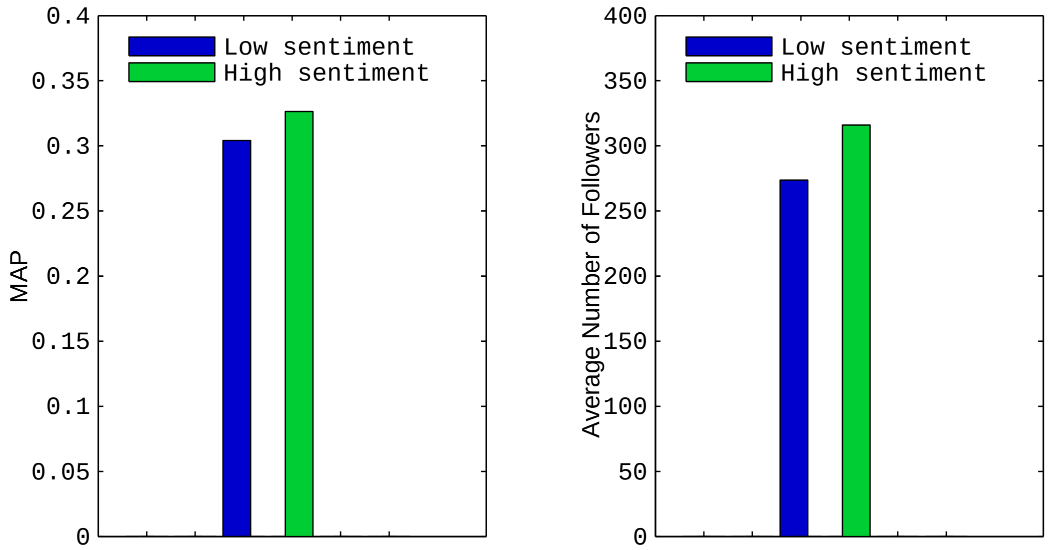


Figure 6.9: The distribution of the user sentiment scores in the training data for all of the users.

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is the total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of those paths that pass through node v.

We calculated the betweenness centrality of each user using the follower network in our dataset. We want to evaluate how well our recommended target users



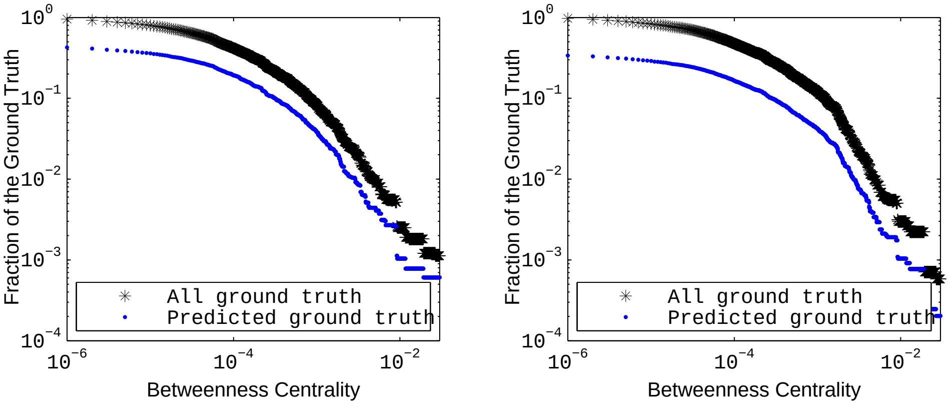
(a) MAP

(b) Average num of followers

Figure 6.10: Comparison of the focal users with the user sentiment scores less than and greater than the average user sentiment score of the focal users for retweet prediction.

are also users with a high betweenness centrality, so that our recommendations are more valuable.

Figure 6.11 reports on the follower network betweenness centrality distribution of all the ground truth users and the subset that can be predicted by our system. The figures were drawn on the log-log scale and the distribution is somewhat close to a power law distribution. The range of the betweenness centrality values for all users in the network is [0,0.06553]. First, we consider users with low betweenness centrality in the range of [0,0.00002). While 28.9 percent of the retweet ground truth and 24.3 percent of the mention ground truth is in that range, the corresponding values for our target recommendations are 27.7% and 21.7% respectively, i.e., we



(a) Retweet Ground Truth (b) Mention Ground Truth

Figure 6.11: Follower network betweenness centrality distribution of the ground truth users. X axis is the follower network betweenness centrality values; Y axis is the fraction of the users that have betweenness centrality values greater than or equal to the corresponding value on X axis. The upper darker distribution represents all ground truth; the lower lighter distribution is the predicted ground truth.

make a lower fraction of our recommendations in this range.

When the betweenness centrality value increases, the curve of the predicted ground truth is closer to the curve of all ground truth for both retweet prediction and mention prediction. Thus, when we consider users with high betweenness centrality, in the range (0.001,0.06553], we see the opposite effect. A higher fraction of target recommendations is in that range. While 8.2 percent of the retweet ground truth and 12.0 percent of the mention ground truth is in that range, we recommend 8.6% and 12.4% respectively. To summarize, we are successful in recommending users in the retweet and mention ground truth that have a higher betweenness centrality.

Chapter 7

Conclusions

In this thesis we address the problem of prediction in social media to select social media channels for monitoring and recommendation. Our analysis focuses on individual authors and linkers. We address a series of prediction problems including future author prediction problem and future link prediction problem in the blogosphere, as well as prediction in microblogs such as twitter. We also study several cases of recommendations based on the prediction work.

For the *Future Author Prediction Problem* in the blogosphere, we develop prediction methods inspired by information retrieval approaches that use historical posts in the blog channel for prediction. We also train a ranking support vector machine (SVM) to solve the problem. We evaluate our methods on an extensive social media dataset; despite the difficulty of the task, all methods perform reasonably well. Results show that ranking SVM prediction can exploit blog channel and diffusion characteristics to improve prediction accuracy. We also found that consistency, diffusion stage (*cRatio*), and blog volume versus author count (V/AC) all impact prediction accuracy. Prediction accuracy increases for consistent blog channels, and with regards to diffusion stage, prediction accuracy is better in the middle stage than in the emerging stage and the declining stage. Prediction accuracy is higher when the V/AC values of the query posts are higher. Although *cRatio* and V/AC

themselves may contain future information, estimates of their current values could be inferred from the historical data. In the situation where diffusion and blog factors can not be controlled, they can still be used to indicate a confidence level for the prediction accuracy of a given query post and provide additional information for recommendation.

An essential element of social media, particularly blogs, is the hyperlink graph that connects various content pieces. There are two types of links within the blogosphere; one from blog post to blog post, and another from blog post to blog channel (an event stream of blog posts). These links can be viewed as a proxy to indicate the flow of information between blog channels and to reflect influence. Given this assumption about links, the ability to predict future links can facilitate the monitoring of information diffusion, making recommendations, and it can improve word-of-mouth (WOM) marketing. For the future link prediction in the blogosphere, we compare multiple link prediction methods, and show that our proposed solution which combines the network properties of the blog with content properties does better than methods which examine network properties or content properties in isolation. Most of the previous work has only looked at either one or the other.

Microblogs such as Twitter support a rich variety of user interactions using tweets, hashtags, urls, retweets and mentions. Microblogs are also an exemplar of a hybrid network; there is an explicit network of followers, as well as an implicit network of users who retweet other users, and users who mention other users. These networks are important proxies for influence. Previous research on diffusion and influence typically assumed that the network was homogeneous. The models were also applied at the aggregate level. We study influence at the level of the individual. We choose a focal user and predict those users who will retweet and/or mention the focal user, in the near future. We use these predictions to make personalized recommendations for applications such as brand monitoring and management. We define a potential function, based on a hybrid network, which reflects the likelihood of a candidate user having a specific type of link in the future to a focal user. We formalize this prediction problem in the hybrid network as an optimization problem (using maximum likelihood). We propose several heuristic solutions that approximate the optimization problem. We perform an extensive evaluation over a microblog network and a stream of tweets from Twitter. Our solutions outperform the baseline methods which only consider one network or naively utilize the hybrid network. The improvement is especially significant for prediction of novel retweeters and novel mentioners where the prediction is more difficult.

We also study the recommendations based on the prediction in the blogoshpere and microblogs. The recommendation system in the blogoshpere we have proposed can provide recommendations that are (1) from Novel Linkers and Authors, (2) diverse, and (3) from blog authors who will both write and link to the focal topic. A brand manager can use this system by feeding it a blog post that is part of a conversation they would like to follow. Then they can use the FAP and FLP predictions to identify new authors and linkers they should monitor. Moreover, the diverse set of recommendations will provide the brand managers with several points of view. Finally, combined recommendations are useful since they identify authors who will post on the topic and link to a focal blog channel. For the recommendation for microblogs, we make recommendations of future retweet and future mention users. We show that sentiment of the focal user appears to have impact on the prediction accuracy and a larger follower network typically reduces the accuracy of our predictions. Our recommendations target future ground truth users with high betweenness centrality values. Those users are potentially more influential. The reason that we are able to identify users with high betweenness centrality values is because our solution is based on a composite network. The users with high centrality values are more likely to receive all tweets in the system; this increases their likelihood of appearing in both the mention and retweet network. Thus, our prediction method that exploits the hybrid network is more likely to identify these more influential users.

Bibliography

[1] Google blog search. http://blogsearch.google.com.

[2] Yahoo! buzz. http://buzz.yahoo.com.

[3] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.

[4] Réka Albert, Bhaskar DasGupta, Riccardo Dondi, Sema Kachalo, Eduardo D. Sontag, Alexander Zelikovsky, and Kelly Westbrooks. A novel method for signal transduction network inference from indirect experimental evidence. *Journal of Computational Biology*, 14(7):927–949, 2007.

[5] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, July 2000.

[6] James Allan, Ron Papka, and Victor Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*, pages 37–45. ACM, 1998.

[7] Eytan Bakshy, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 65–74, 2011.

[8] Nilesh Bansal and Nick Koudas. Blogscope: A system for online analysis of high volum text streams. In *Proceedings of the International Conference on Very Large data Bases (VLDB)*, 2007.

[9] Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 36–44, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[10] Frank M. Bass. A new product growth for model consumer durables. *Management Science*, 15(5):215–227, 1969.

[11] Hila Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: Real-world event identification on twitter. In *ICWSM '11: Proceedings of the International Conference on Weblogs and Social Media*, 2011.

[12] Albert Bifet and Eibe Frank. Sentiment knowledge discovery in twitter streaming data. In *Proceedings of the 13th international conference on Discovery science*, DS'10, pages 1–15, Berlin, Heidelberg, 2010. Springer-Verlag.

[13] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[14] BlogPulse. http://www.blogpulse.com.

[15] Phillip Bonacich. Power and centrality: A family of measures. *The American Journal of Sociology*, 92(5):1170–1182, 1987.

[16] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 89–96, 2005.

[17] K. Burton, A. Java, and I. Soboroff. The icwsm 2009 spinn3r dataset. In *Proceedings of the Conference on Weblogs and Social Media (ICWSM 2009)*, 2009.

[18] Deng Cai, Qiaozhu Mei, Jiawei Han, and Chengxiang Zhai. Modeling hidden topics on document manifold. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 911–920, 2008.

[19] Deng Cai, Xuanhui Wang, and Xiaofei He. Probabilistic dyadic data analysis with local and global consistency. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 105–112, 2009.

[20] Damon Centola. The spread of behavior in an online social network experiment. *Science*, 329(5996):1194–1197, 2010.

[21] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and Krishna Gummadi. Measuring user influence in twitter: The million follower fallacy. In *in ICWSM '10: Proceedings of international AAAI Conference on Weblogs and Social*, 2010.

[22] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SigKDD)*, 2010.

[23] J. Chevalier and D. Mayzlin. The effect of word of mouth online: online book reviews. *Journal of Marketing Research*, 43:348–354, 2006.

[24] William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things. In *Proceedings of the 1997 conference on Advances in neural information processing systems 10*, NIPS '97, pages 451–457, Cambridge, MA, USA, 1998. MIT Press.

[25] Elizabeth M. Daly and Mads Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '07, pages 32–40, New York, NY, USA, 2007. ACM.

[26] Abhinandan S. Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 271–280, New York, NY, USA, 2007. ACM.

[27] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 241–249, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[28] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[29] Minghua Deng, Shipra Mehta, Fengzhu Sun, and Ting Chen. Inferring domain-domain interactions from protein-protein interactions. In *Proceedings of the sixth annual international conference on Computational biology*, RECOMB '02, pages 117–126, 2002.

[30] Christopher P. Diehl, Galileo Namata, and Lise Getoor. Relationship identification for social network discovery. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 1*, AAAI'07, pages 546–552, 2007.

[31] Digg. http://digg.com.

[32] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66, New York, NY, USA, 2001. ACM.

[33] Khalid El-Arini, Gaurav Veda, Dafna Shahaf, and Carlos Guestrin. Turning down the noise in the blogosphere. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 289–298, 2009.

[34] W. L. Ferrar. Finite matrices. *Oxford Univ. Press*, 1951.

[35] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, December 2003.

[36] Minas Gjoka, Maciej Kurant, Carter T. Butts, and Athina Markopoulou. Walking in facebook: a case study of unbiased sampling of osns. In *Proceedings of the 29th conference on Information communications*, INFOCOM'10, pages 2498–2506, 2010.

[37] David Godes and Dina Mayzlin. Firm-created word-of-mouth communication: Evidence from a field test. *Marketing Science*, 28(4):721–739, 2009.

[38] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12(3):211–223, 2001.

[39] R. Grewal, T. Cline, and A. Davies. Early-entrant advantage, word-of-mouth communication, brand similarity, and the consumer decision-making process. In *Journal of Consumer Psychology*, volume 13, pages 187–197, 2003.

[40] Daniel Gruhl, R. Guha, David Liben-Nowell, and Andrew Tomkins. Information diffusion through blogspace. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 491–501, New York, NY, USA, 2004. ACM.

[41] John Hannon, Mike Bennett, and Barry Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 199–206, 2010.

[42] David Heckerman, David Maxwell Chickering, Christopher Meek, Robert Rounthwaite, and Carl Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *J. Mach. Learn. Res.*, 1:49–75, September 2001.

[43] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, pages 115–132, Cambridge, MA, 2000. MIT Press.

[44] William Hersh, Chris Buckley, T. J. Leone, and David Hickam. Ohsumed: an interactive retrieval evaluation and new large test collection for research. In *Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval*, SIGIR '94, pages 192–201, 1994.

[45] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 50–57, 1999.

[46] Hailiang Huang and Joel S. Bader. Precision and recall estimates for two-hybrid screens. *Bioinformatics*, 25(3):372–378, February 2009.

[47] Seungil Huh and Stephen E. Fienberg. Discriminative topic modeling based on manifold learning. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 653–662, 2010.

[48] Pan Hui, Augustin Chaintreau, James Scott, Richard Gass, Jon Crowcroft, and Christophe Diot. Pocket switched networks and human mobility in conference environments. In *Proceedings of the 2005 ACM SIGCOMM workshop*

on Delay-tolerant networking, WDTN '05, pages 244–251, New York, NY, USA, 2005. ACM.

[49] Pan Hui, Jon Crowcroft, and Eiko Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '08, pages 241–250, New York, NY, USA, 2008. ACM.

[50] Takahiko Ito, Masashi Shimbo, Taku Kudo, and Yuji Matsumoto. Application of kernels to link analysis. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 586–592, New York, NY, USA, 2005. ACM.

[51] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 133–142, New York, NY, USA, 2002. ACM.

[52] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18:39–40, 1953.

[53] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, New York, NY, USA, 2003. ACM.

[54] Jon Kleinberg. Computing: The wireless epidemic. *Nature*, 449(7160):287–288, 2007.

[55] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 426–434, 2008.

[56] Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. Twitter sentiment analysis: The good the bad and the omg! In *in ICWSM '11: Proceedings of international AAAI Conference on Weblogs and Social*, 2011.

[57] Jérôme Kunegis and Andreas Lommatzsch. Learning spectral graph transformations for link prediction. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 561–568. ACM, 2009.

[58] Jen-Wei Kuo, Pu-Jen Cheng, and Hsin-Min Wang. Learning to rank from bayesian decision inference. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 827–836, 2009.

[59] Vincent Leroy, B. Barla Cambazoglu, and Francesco Bonchi. Cold start link prediction. In *KDD '10: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 393–402, New York, NY, USA, 2010. ACM.

[60] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SigKDD)*, 2009.

[61] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 631–636, 2006.

[62] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 641–650, New York, NY, USA, 2010. ACM.

[63] C. Li and J. Bernoff. Groundswell: Winning in a world transformed by social technologies. *Harvard Business School Press*, 2008.

[64] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 556–559, New York, NY, USA, 2003. ACM.

[65] Cindy Xide Lin, Bo Zhao, Qiaozhu Mei, and Jiawei Han. Pet: a statistical model for popular events tracking in social communities. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 929–938, 2010.

[66] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

[67] Nathan N. Liu, Min Zhao, Evan Xiang, and Qiang Yang. Online evolutionary collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 95–102, 2010.

[68] Gang Luo, Chunqiang Tang, and Philip S. Yu. Resource-adaptive real-time new event detection. In *Proceedings of the SIGMOD international conference on Management of data*, pages 497–508, 2007.

[69] Christopher Manning, Prabhakar Raghavan, and Hinrich Schutze. *Introduction to Information Retrieval.* Cambridge University Press, 2008.

[70] Liam McNamara, Cecilia Mascolo, and Licia Capra. Media sharing based on colocation prediction in urban transport. In *Proceedings of the 14th ACM*

international conference on Mobile computing and networking, MobiCom '08, pages 58–69, New York, NY, USA, 2008. ACM.

[71] Derek Monner. Tweet sentiment computation. *National Science Foundation SM3 Project Wiki, Smith School of Business*, 2011.

[72] Shishir Nagaraja and Ross Anderson. The topology of covert conflict. In *WEIS*, 2006.

[73] Galileo Mark Namata, Hossam Sharara, and Lise Getoor. A survey of link mining tasks for analyzing noisy and incomplete networks. In Jiawei Han Philip S. S. Yu and Christos Faloutsos, editors, *Link Mining: Models, Algorithms, and Applications*. Springer, 2010.

[74] M. E. J. Newman. Clustering and preferential attachment in growing networks. *Phys. Rev. E*, 64(2):025102, Jul 2001.

[75] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *in ICWSM '10: Proceedings of international AAAI Conference on Weblogs and Social*, 2010.

[76] Joshua O'Madadhain, Jon Hutchins, and Padhraic Smyth. Prediction and ranking algorithms for event-based network data. *SIGKDD Explor. Newsl.*, 7(2):23–30, December 2005.

[77] Iadh Ounis, Craig Macdonald, and Ian Soboroff. Overview of the trec-2008 blog track. In *Proceedings of the Text REtrieval Conference (TREC)*, 2008.

[78] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*, 2010.

[79] Wei Pan, Nadav Aharony, and Alex Pentland. Composite social network for predicting mobile apps installation. In *Twenty-Fifth Conference on Artificial Intelligence (AAAI)*, 2011.

[80] M. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27:313–331, 1997.

[81] Saša Petrović, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–189, Los Angeles, California, June 2010. Association for Computational Linguistics.

[82] Jonathan Reades, Francesco Calabrese, Andres Sevtsuk, and Carlo Ratti. Cellular census: Explorations in urban data collection. *IEEE Pervasive Computing*, 6(3):30–38, July 2007.

[83] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 61–70, New York, NY, USA, 2002. ACM.

[84] S.E. Robertson, S. Walker, and M. Hancock-Beaulieu. Okapi at trec-7: automatic ad hoc, filtering, vlc and interactive track. *TREC 1998*, pages 199–210, 1998.

[85] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In *TREC-3*, pages 109–126, 1994.

[86] Haggai Roitman, David Carmel, and Elad Yom-Tov. Maintaining dynamic channel profiles on the web. *Proc. VLDB Endow.*, 1(1):151–162, 2008.

[87] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 791–798, 2007.

[88] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., New York, NY, USA, 1986.

[89] Hassan Sayyadi, Matthew Hurst, and Alexey Maykov. Event detection and tracking in social streams. In *Proceedings of International Conference on Weblogs and Social Media (ICWSM)*, 2009.

[90] Amit Singhal. Modern information retrieval: a brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24(4):35–43, 2001.

[91] Han Hee Song, Tae Won Cho, Vacha Dave, Yin Zhang, and Lili Qiu. Scalable proximity estimation and link prediction in online social networks. In *IMC '09: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 322–335, New York, NY, USA, 2009. ACM.

[92] Libo Song, Udayan Deshpande, Ulas C. Kozat, David Kotz, and Ravi Jain. Predictability of wlan mobility and its effects on bandwidth provisioning. In *INFOCOM*, 2006.

[93] Xiaodan Song, Yun Chi, Koji Hino, and Belle L. Tseng. Information flow modeling based on diffusion rate for prediction and ranking. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 191–200, New York, NY, USA, 2007. ACM.

[94] Neil Spring, David Wetherall, and Thomas Anderson. Reverse engineering the internet. *SIGCOMM Comput. Commun. Rev.*, 34(1):3–8, January 2004.

[95] Karthik Subbian and Prem Melville. Supervised rank aggregation for predicting influencers in twitter. In *SocialCom/PASSAT*, pages 661–665, 2011.

[96] Prem Swaroop, Yogesh Joshi, William Rand, and Louiqa Raschid. Modeling behavior and its effect on diffusion effectiveness in microblogging. *Technical Report, Smith School of Business*, 2012.

[97] Andras Szilagyi, Vera Grimm, Adrin K Arakaki, and Jeffrey Skolnick. Prediction of physical proteincprotein interactions. *Physical Biology*, 2(2):S1, 2005.

[98] J. Tang, C. Mascolo, M. Musolesi, and V. Latora. Exploiting temporal complex network metrics in mobile malware containment. In *Proceedings of the 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, WOWMOM '11, pages 1–9, Washington, DC, USA, 2011. IEEE Computer Society.

[99] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 807–816, New York, NY, USA, 2009. ACM.

[100] Benjamin Taskar, Ming-Fai Wong, Pieter Abbeel, and Daphne Koller. Link prediction in relational data. In *Proceedings of Neural Information Processing Systems (NIPS)*, 2003.

[101] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *in ICWSM '10: Proceedings of international AAAI Conference on Weblogs and Social*, 2010.

[102] Iraklis Varlamis, Vasilis Vassalos, and Antonis Palaios. Monitoring the evolution of interests in the blogosphere. In *IEEE ICDE Workshops*, pages 513–518, 2008.

[103] Chao Wang, Venu Satuluri, and Srinivasan Parthasarathy. Local probabilistic models for link prediction. In *ICDM '07: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 322–331, Washington, DC, USA, 2007. IEEE Computer Society.

[104] Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 448–456, 2011.

[105] Jianshu Weng and Bu-Sung Lee. Event detection in twitter. In *ICWSM '11: Proceedings of the International Conference on Weblogs and Social Media*, 2011.

[106] Frank Wilcoxon. Individual comparisons by ranking methods. In *Biometrics Bulletin*, volume 1, pages 80–83, December 1945.

[107] Shanchan Wu, Tamer Elsayed, William Rand, and Louiqa Raschid. Predicting author blog channels with high value future posts for monitoring. In *Twenty-Fifth Conference on Artificial Intelligence (AAAI)*, 2011.

[108] Shanchan Wu, Leanna Gong, William Rand, and Louiqa Raschid. Making recommendations in a microblog to improve the impact of a focal user. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '12, 2012.

[109] Shanchan Wu, William Rand, and Louiqa Raschid. Recommendations in social media for brand monitoring. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, pages 345–348, 2011.

[110] Shanchan Wu, Louiqa Raschid, and William Rand. Future link prediction in the blogosphere for recommendation. In *ICWSM '11: Proceedings of the International Conference on Weblogs and Social Media*, 2011.

[111] Jun Xu and Hang Li. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 391–398, 2007.

[112] Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and on-line event detection. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 28–36, New York, NY, USA, 1998. ACM.

[113] Yin Yang, Nilesh Bansal, Wisam Dakka, Panagiotis Iperiotis, Nick Koudas, and Dimitris Papadias. Query by document. In *Proceedings of WSDM*, 2009.

[114] Haiyuan Yu, Alberto Paccanaro, Valery Trifonov, and Mark Gerstein. Predicting interactions in protein networks by completing defective cliques. *Bioinformatics*, 22(7):823–829, April 2006.

[115] Elena Zheleva, Lise Getoor, Jennifer Golbeck, and Ugur Kuter. Using friendship ties and family circles for link prediction. In *Proceedings of the Second international conference on Advances in social network mining and analysis*, SNAKDD'08, pages 97–113, 2010.