The Pennsylvania State University

The Graduate School

MINING USER-GENERATED CONTENTS ON THE WEB AND SOCIAL NETWORKS

A Dissertation in

Information Science and Technology

by

Shu Huang

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

August 2013

The thesis of Shu Huang was reviewed and approved* by the following:

Dongwon Lee
Associate Professor of Information Sciences and Technology
Thesis Advisor, Chair of Committee

Peng Liu
Professor of Information Sciences and Technology

Heng Xu
Associate Professor of Information Sciences and Technology

Jack Hayya
Professor Emeritus of Supply Chain and Information System

Mary Beth Rosson
Professor of Information Sciences and Technology
Director of Graduate Programs

*Signatures are on file in the Graduate School.

Abstract

In solving diverse data management problems, underlying social network between users and semantics hidden deep in User-generated Contents (UGC) can be useful from many perspectives. Finding and applying such hidden semantics of UGC and social correlations illustrates a new way in solving various problems. In this thesis, we study several challenging data management problems to investigate how to apply the framework of UGC mining and social network analysis to substantially improve existing solutions. In particular, we focus on the following four problems:

First, we propose a novel query expansion technique in Information Retrieval that exploits the "location-based" correlation between users and search engine user logs. We explore the vocabulary of users from different geographic locations and investigate the semantic relations among the documents they search for. Based on that, a hierarchical location and topic based query expansion model is proposed to improve the accuracy of web search. Our proposed model predicts the query location sensitivity with more than 80% precision. Using the model, the final search result is significantly better than several existing query expansion methods.

Second, we explore the aggregate social activity and evaluate the significance of

various activity features in determining the social activity evolution. In particular, we look in to various formats of social activities and measure how member activity impacts the evolution of the active population. Several activity features are extracted and their impact on the community evolution is evaluated with a feature selection model. Based on the model, the most significant features are identified.

Third, we study UGC on Twitter, a large online platform of social media, to identify tweet topics and sentiments towards some preset brands/products. To help understand brand perception and customer opinions, we utilize the correlation of tweet sentiments and topics, and propose a multi-task multi-label (MTML) classification model that performs classification of both sentiments and topics simultaneously. It incorporates results of each task from prior steps to promote and reinforce the other iteratively. Meanwhile, by using multiple labels, the class ambiguity can be addressed. Compared with baselines, MTML produces a much higher accuracy of both sentiment and topic classification.

Furthermore, based on tweet sentiment analysis, social network among Twitter users is also taken into consideration to investigate the impact of events on tweet sentiment change. By mining tweets about 2012 USA presidential campaign, we analyze the sentiments towards the presidential candidates. Meanwhile, we incorporate social correlation between Twitter users and present a method to predict the impact of events based on social activities. Analysis on tweets collected over 8 months shows that our method can predict the sentiment change with high accuracy. Mining UGC and social network is not only efficient but also effective in predicting the impact of events.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The studies on mining user generated contents(UGC) and interpretation have been rapidly evolving in recent years. With the emergence of digital storage and a variety of online services, sources arise to enable in-depth research in relevant applications in various data management problems. In this dissertation, the semantic computing framework is adopted to learn from UGC and improve techniques in information retrieval, social activity mining, and sentiment analysis as well as prediction in social media.

UGC has a variety of formats, which are supported by different social media websites, such as Facebook, Tumblr, Twitter, and Youtube. Figure 1.1 shows an example of UGC from Twitter. On these websites, users can write and post UGC about any subject. At the same time, a user can also follow other users, which forms online social network. Therefore any update of the followed users will be delivered to the follower. On one hand, social network determines the diffusion of UGC, thus analyzing social network structure can help predict the generation of UGC. On the

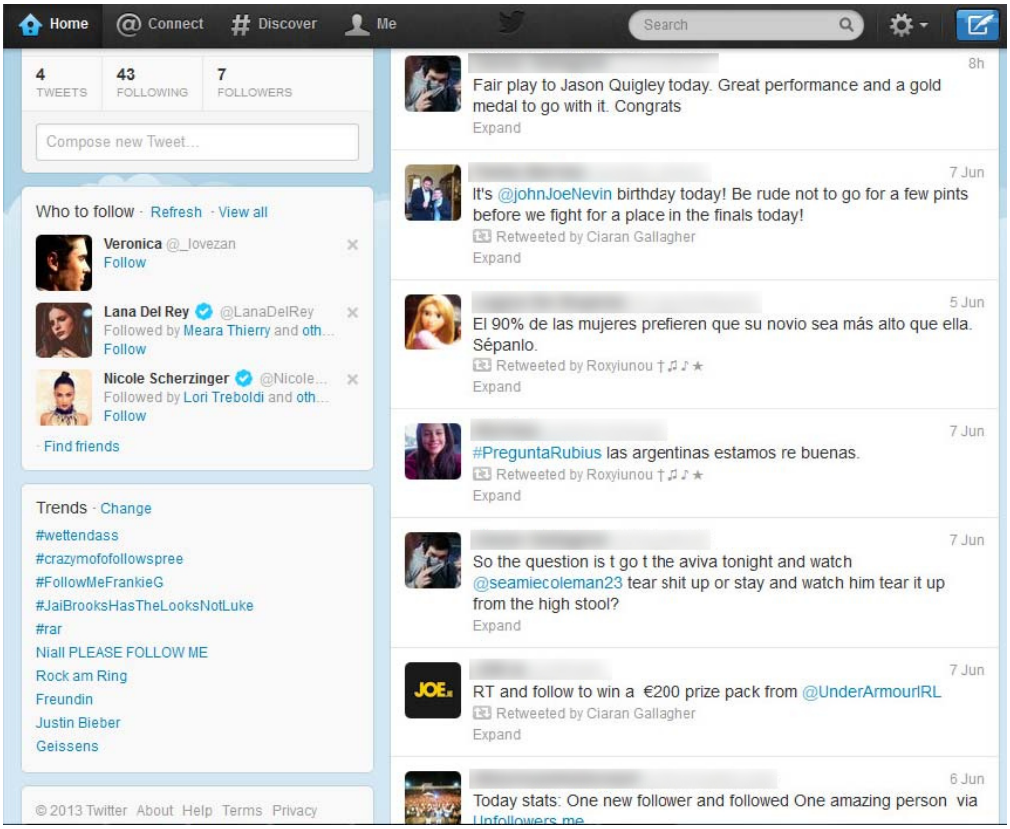


Figure 1.1. An example of user generated contents: twitter stream

other hand, UGC also have impacts on individuals, which furthermore promotes the change of social network topology. Studying UGC and social connections can provide insights to this interactive relationship. Mining and understanding the correlations will help with solving a lot of problems.

As a study of interpretation analysis, UGC mining focuses on analyzing words, signs and symbols, as well as understanding their meaning. Based on that, semantic computing on UGC explores hidden semantics in UGC content, such as topics and sentiments. Semantic relations embodied in UGC reveal deep patterns and domain rules in a variety of fields. The objective of semantic computing on UGC is to understand the meaning of various sorts of computational content and furthermore find out mapping rules.

Generally, the topics addressed in mining UGC and semantic computing can be grouped into three categories: intention understanding, content interpretation, and semantic mapping. The first category involves understanding the intentions of human expression and convert them into a machine-processable language. In the second category, the focus is on understanding and converting various sorts of user generated content, including but not limited to text, video, audio, and image. Semantic mapping is based on the results of the first two categories. By understanding semantic objectives and extracting their relations, algorithms are developed to create mapping between semantic content for different purposes.

As the relations generated with semantic mapping are integrated, the meaning of semantic objectives can be understood within a common framework. Thus, relevant patterns for different purposes can be summarized from the computational content through the embedded semantic metadata. With the patterns extracted, many methods are proposed to apply UGC mining to different applications, including document parsing, semantic relation extraction, semantic interpretation, and entity disambiguation.

Existing researches have developed many techniques of semantics analysis from different perspectives. To study the semantic relations, Bollegala et al. proposed a relational similarity measure to compute the similarity between semantic relations by using a web search engine [1]. Furthermore, they presented a clustering algorithm to train the logistic regression model to identify the relation patterns expressed by each cluster [2]. By using a wide-coverage parser and semantic analyzer trained from newspaper text, a method is proposed to generate large scale semantic knowledge networks efficiently [3]. Another technique is proposed for scalable semantic retrieval by adopting summarization and refinement [4]. Based

on visual and semantic consistency, a social image retagging scheme is developed to assign images with better content description [5]. These studies analyze the semantic metadata in text and summarize the mapping rules for further analysis.

In data management, several studies are conducted to explore applications of semantic mining on the web and social networks. In [6], a technique is introduced to perform large scale semantic integration and reduce the redundancy while indexing on semantic web. Also, a decentralized infrastructure is proposed to efficiently tackle the graph-based entity disambiguation problem [7]. As an enhanced application, a semantic web search engine Falconer is developed to support friends auto-discovery, semantic annotation, as well as topic trend analysis [8]. Furthermore, by projecting each post into a topic space, a sparse coding-based model is constructed in [9] to simultaneously model semantics and structure of threaded discussions. In a variety of applications, relevant problems are addressed by mining semantics from UGC and utilizing the patterns extracted.

In this dissertation, we make use of UGC mining to help solve data management problems in a few domains, including query expansion in web search, social activity analysis, multi-label sentiment and topic co-classification, and prediction on impact of events. The UGC involved includes not only search engine user logs, but also online posts and microblogs in social networks.

Chapter 2

Background and Problem Statement

2.1 Overview

Exploiting semantics in UGC, we study four problems in domains of information retrieval, social activity analysis, topic and sentiment classification, and prediction of impact of events. The UGC mining techniques are applied to propose new methodologies or substantially improve existing solutions. In this chapter, we introduce the background of each problem and present the problem statement.

Search engine user log is a type of adhoc user generated contents in web search. By mining search engine user log, we can explore not only the geographic correlation between users, but also semantics of searching objectives. In this dissertation, we make use of user location and semantic topic of searching objective to develop a new query expansion method to improve web search.

After that, we study how user interactions in a social network have influence on growth or shrinkage of social activity. By applying lasso-based logistic regression, we find out the most significant features that determine aggregate social activity evolution on two different types of social networks.

Since social activity and UGC are both dynamically evolving and highly correlated, investigating the interactive relationship between them can provide insights that will promote predictions on social networks. Among all aspects, understanding sentiments and topics of UGC has a wide application in online marketing and advertising. For instance, it helps business owners monitor the customer opinions and attitude towards a brand/product. Therefore, we analyze Twitter stream and propose a multi-task multi-label classification model that can classify tweet sentiments and topics simultaneously and efficiently . Furthermore, by integrating social activity and UGC sentiment, we analyze the temporal change of sentiment in Twitter stream. Based on aggregate social activity, we present a method that can simulate and predict sentiment change of tweets over time.

2.2 Query Expansion in Information Retrieval

The objective of information retrieval is to provide relevant information to different users out of the overwhelming amount of data according to their searching keywords/queries. Based on the observation that users often issue very short queries, query expansion techniques have been proposed to close the gap between brief expressions and retrieval objectives.

Query expansion is the process that reformulate a seed query to improve retrieval performance in information retrieval [10]. Within short queries, the same keywords may be interpreted into different topics for different users, which may also be true for the same users. The basic idea behind query expansion is to add extra keywords to the short queries so that the retrieval objective can be expressed more specifically and accurately. In this way, more accurate retrieval results can

be obtained [11].

Different query expansion techniques proposed in past years [12] [13] can be generally grouped into three categories: document-based query expansion, term-based query expansion, and concept-based query expansion. We observe that the existing methods ignored two important issues. First, users from different locations may have different vocabularies and hence they may refer to the same objects with different query terms, which we identify as *query location sensitivity*. For example, in British English, the term *lorry* represents what *truck* refers to in American English. Also, the term *paddock* has different meanings in Australian English and British English. As a result, documents on the same topic created by users from different locations differ in their lists of keywords, while the same thing also happens on the queries issued by users from different locations. Therefore, the distribution of keywords in queries will reflect the term usage distribution. Second, the same search keywords may refer to different topics by different users under different context.

The search engine user log file records the semantic relation of searching keywords and web documents. Therefore, we propose to investigate the user log and combine the semantic topics with location information into an efficient methodology for query expansion. First, we show by experiments that some of the queries are location-sensitive and others not. Second, for location-sensitive queries, we present two types of expansion strategies: same-location based query expansion and different-location based query expansion. We also propose a hierarchical classification model to classify a new query into different types at two levels (location sensitive versus location non-sensitive, then same location sensitive versus different location sensitive).

While analyzing the search engine log data, IP addresses are used to locate users, and derived topics of each document are used to represent the query objectives. Rather than deterministic document classification, we utilize the Latent Dirichlet Allocation (LDA) model [14] in which each document is represented as a vector of semantic topics. The similarity of two documents is calculated by the corresponding vectors. Furthermore, the similarity of two queries is determined by comparing the two sets of clicked documents therein. Last, keywords in top similar queries are added to the initial query as expansion.

2.3 Temporal Social Activity Analysis

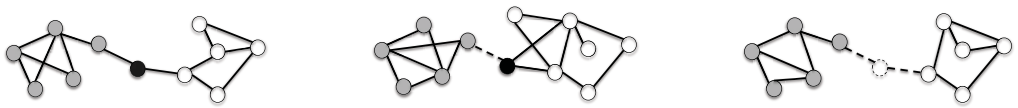
Intensive communication between individuals leads to complex social network infrastructures and a huge amount of UGC. With the emergence of online communities, many sources arise to enable in-depth research in social network analysis. Exploring the social network evolution helps people understand how the community evolves over time [15]. Based on that, the evolution of UGC can be investigated accordingly.

Within a social network, members interact and produce UGC frequently, such as videos, pictures, blogs, and tags. Parsing and extracting semantic topics embodied through UGC reveal the topic trend and patterns of the topic correlation. At the same time, semantic relations of UGC also indicate the latent relationship between the corresponding individuals, e.g. common interest in the same objectives. With the interactive population forming dynamic communities, the change of semantic topic distribution can be explored by studying the community evolution.

Among all factors, member interaction plays a most important role in determining the community evolution. New members are usually introduced into the community by the existing active members, instead of the inactive ones. On the other hand, the active population not only determines the social network evolution but also reveals the business value of the community. For instance, if the members have frequent activities and the active population keeps increasing, then it is worthwhile to invest in advertisements and promotions for this community and vice versa. In businesses, such as advertising, ex-ante knowledge of the status of the targeted community will assist the decision making of the advertisers and therefore help improve the profit.

Different methodologies are proposed to analyze and predict the social network evolution. Generally, they can be grouped into static network mining, microscopic evolution prediction, and structure analysis [16, 17, 18]. Observing existing work, we find that they overlook three important issues. First, cumulative social network does not emphasize the current member interaction; thus, the inactive members are also included in predicting the network evolution. Example 1 illustrates the biased impact by including inactive members when measuring a social network. Second, the members with the same count of existing connections may have different activities. Thus the inactive members may not have the same probability as the active ones in attracting new attachment. Third, structural features not only measure the network status but also provide a good summary of the member interaction. Therefore, they can be adopted to help with the prediction of the social network evolution.

Example 1 (Member Activity). As a motivating example, consider a social network where members register first and then begin the involvement in diverse



(a)Member activity at t_1 (b)Member activity at t_2 (c)Member activity at t_3

Figure 2.1. Illustration of the impact of inactive members in a social network: (a) original network with two groups bridged by the black node; (b) the black node becomes inactive toward one group (represented as the dotted edge); and (c) the black node becomes completely inactive, separating two once-connected groups.

activities at various groups. Figure 2.1(a) depicts the member activities in such a social network with two groups (i.e. the set of grey and white nodes) at time t_1. The groups are connected via the black node v_b. After some time, as shown in Figure 2.1(b), v_b becomes dormant toward the grey node group. Then, although a cumulative network during interval $[t_1, t_3]$ would still show that two groups are connected via v_b, truth is that the structure and characteristics of the network has been altered due to the change of activity of v_b. Finally, in Figure 2.1(c), v_b is no longer active and becomes completely isolated in the network. □

In this dissertation, we focus on investigating the active community evolution. To study the active population, first, we present an approach to evaluate member activities and measure their impact on community evolution. Also we introduce a model that incorporates the structural features to predict the network evolution quantitatively. At the same time, a shrinkage method is adopted to find the most significant structural features in determining the evolution. At last, we show by experiment that using the selected most valuable features can improve prediction accuracy.

2.4 Sentiment and Topic Classification in Social Media

As online social network services become more popular in recent years, microblogging such as Twitter, has been rapidly growing. Users post short texts, called *tweets*, about any topic of interest, reply to others' tweets, and disseminate information to other users by re-tweeting. Although tweets are limited to no more than 140 characters, Twitter has become an extremely popular platform where people freely express and exchange opinions.

Businesses in particular has noticed the potential of Twitter and used it in a variety of applications, such as marketing promotion, brand campaign, and customer care [19]. For instance, a lot of companies have started to poll relevant tweets to help understand trending topics among their customers and the sentiments towards their products.

Among all knowledge that can be extracted from tweets, in this dissertation, we focus on two aspect: (1) *sentiment* of a tweet that captures the subjective mood of a user, such as "positive" and "negative"; and (2) *topic* of a tweet that indicates the scope of subject content from pre-determined aspects, such as "Compliment", "News", and "Promotion". In general, techniques known as *sentiment analysis* and *topic analysis* respectively are used to infer latent sentiments and topics of a given text corpus. Furthermore, in this dissertation, we employ the following class schemes. The sentiment classes are "positive", "negative", and "neutral". The topic classes include "Care/Support", "Lead/Referral", "Mention", "Promotion", "Review", "Complaint", "Inquiry/ Question", "Compliment", "News", and "Company/Brand". We focus on the problem of *classification*, i.e., given a set of

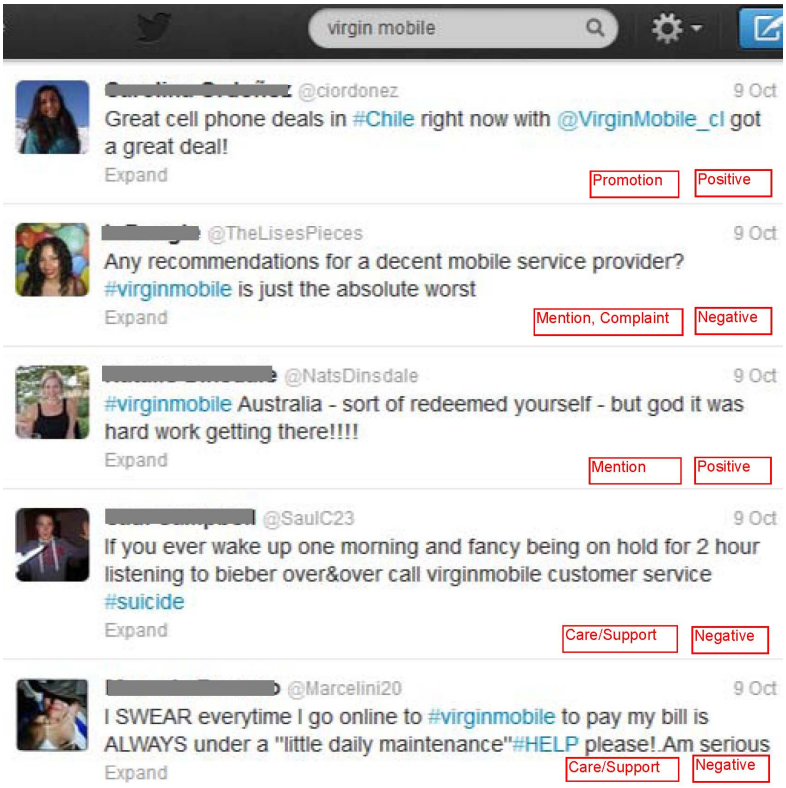


Figure 2.2. Tweets related to "virgin mobile", with topic and sentiment labels.

pre-determined classes, how to identify which classes an instance belongs to.

Given a collection of tweets regarding a certain common subject, a topic classification method can reveal the particular aspects that users are talking about and which are dominant, while a sentiment classification method tells the proportion of users who feel positive or negative toward the subject.

For instance, Figure 5.1 shows example tweets related to "virgin mobile", with their identified sentiment and topic labels. In this example, some users are talking about promotions, and others are complaining about customer service and payment. Meanwhile, some tweets show positive sentiment about the brand, while others are negative. As one can see, therefore, the analysis of tweet sentiments and topics can help businesses to get a sense of user opinion towards their prod-

ucts and services. Due to the practical implication, in recent years, a lot of studies (e.g., [20, 21, 22, 19, 23]) have been conducted towards sentiment and topic classifications of tweets (see Section 2 for details).

However, by and large, existing solutions have the following issues. First, conventional solutions usually treat sentiment and topic classification tasks separately, though the two tasks are often closely related. For instance, tweets about some topics usually tend to have certain sentiment. In Figure 5.1, a user who tweets about "promotions" shows positive sentiment, while two other users who complain about the "care/support" appears to be negative. It implies that often tweet topics can help promote the sentiment classification, and vice versa. On the other hand, the same words could present different sentiments in different topics. Therefore, one can exploit such an inter-relationship between two classification tasks to improve the overall classification accuracy. Second, compared to traditional document corpus where sentiment or topic classification occurs, micro-blog data such as tweets are very short, noisy, and ambiguous. For instance, a tweet mentioning a broken mobile device may be assigned to either the topic of "complaint" or "care/support". Therefore, instead of insisting on the assignment of a single class label to a tweet, sometimes, one can flexibly assign multiple class labels to an ambiguous tweet.

Based on the two limitations of existing methods, in this dissertation, we propose a novel model, termed as the *Multi-Task Multi-Label*(MTML), which performs the classification of both sentiments and topics of tweets concurrently, and incorporates each other's results from prior steps to promote and reinforce current results iteratively. The learned class labels of one task are incorporated as part of predicting features of the other task. For each task, the model is trained with the

maximum entropy by using multiple labels to learn more information and handle class ambiguity. In addition, the MTML model produces probabilistic results, instead of binary results, so that multi-label prediction is allowed and labels can be ranked accordingly.

2.5 Predict Sentiment Change in Twitter Stream

As social media grows rapidly in recent years, a lot of studies have been conducted to make prediction on various realms by mining social media. Two popular social media websites, Facebook and Twitter, provide a wide platform where mining UGC can reveal very valuable information, such as users' opinions and sentiments, towards many aspects. By investigating real life events and online social activities, UGC has been used to predict marketing, movie box-office revenue, and political elections [24].

However, most existing work analyzes sentiments from UGC, e.g. tweets, and apply the results in prediction of other realms. Per our best knowledge, no work has yet addressed the prediction of sentiment change in UGC. In this dissertation, by exploring aggregate social activity in twitter social network and analyzing tweet sentiments, we present a novel method that can predict the impact of events, in particular, predict aggregate sentiment change expressed in tweets.

We collected tweets about 2012 USA presidential campaign. The proposed method is applied on this dataset and focuses on the presidential candidates Barack Obama and Mitt Romney. Shown by experiments, we validate our method and predict the temporal sentiment change of twitter users towards the presidential candidates upon their electoral activities.

Chapter 3

Mining Web Logs for Query Expansion

In this chapter, we present a novel query expansion technique in information retrieval. Search engine user log, as a format of UGC that records search keywords and click through actions, provides hidden semantic relations of searching objectives and location information of users. By investigating the semantic topics in searching documents and extracting the mapping rule, a query classification model is created and furthermore the location-based query expansion strategy is developed.

3.1 Literature Survey

Query expansion techniques have shown significant improvements in the effectiveness of information retrieval systems. Existing methods can be categorized into document-based methods, term-based methods and concept-based methods.

Many earlier algorithms with conventional probabilistic retrieval approach are

document-based [25]. With this approach, an initial query is executed and a set of documents are returned. Then a set of terms are obtained from the top relevant documents, which are combined with the initial query to generate and return a more relevant set of documents. Cai, et. al. propose a method based on the divergence of the query, which calculates the relevance of queries according to their distribution in documents [12]. Also probabilistic models, such as Markov Chains, are applied to improve performance by combining different methods at successive stages [26].

In the category of term-based methods, term relationship has been widely used. Synonyms, co-occurrence, and WordNet are integrated into one language model to explore the relevance of terms [27]. Similarly, term relationship and information flow are explored to supplement single terms with term sets [28]. Other methods in [29, 30, 11], though, propose to obtain relevance of queries by mining click-through data, fail to notice other features of the query.

The concept-based methods pay more attention to user interaction [31]. For a short query, the algorithm returns a list of concepts to be selected by the user. After that, the selected concepts are added back to the initial query as expansion.

Different from existing approaches, in this dissertation, we propose a two-level location-based query expansion model. The location-based query expansion is superior to other query expansion approaches on location-sensitive queries. We broke through the document and the term levels, and explored the semantics embedded in the queries at different granularities.

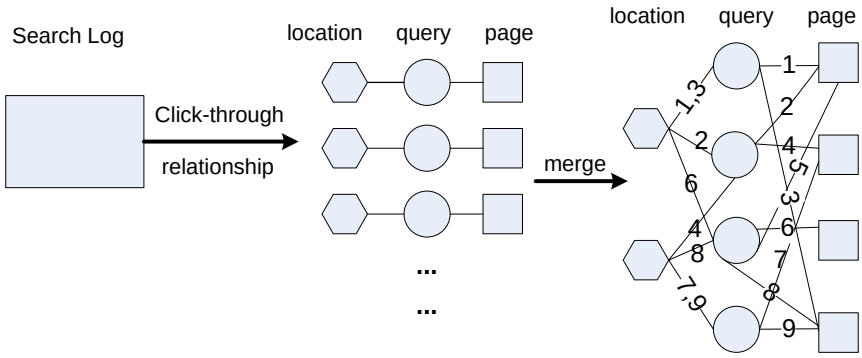


Figure 3.1. Representation of search log data

3.2 Preliminaries

Search engine log data, also called click-through data, keeps the records of interactions between web users and the searching engine. Merging these user sessions, we can construct a triple graph as Figure 3.1 shows. By looking at the user sessions in real click-through data from Citeseer and Excite, we have the following observations:

- Users from different countries issue different queries to represent the same documents.
- Users from different countries use the same queries but with different interpretation.

The above two observations obtained from the log data motivate our research on investigating the location factor in information retrieval and especially in query expansion. Table 3.1 shows some examples from the log data.

Given a query q, the default search results returned by the search engine are represented as Lf, which is a list of ranked documents. Suppose that each document is represented as a vector of topics, in which each entry represents the

Table 3.1. Different queries, different location, same DocID in Excite log data

Query	Country	Excite DocID
wintv	DE	70007
hauppauge	LU	70007
bruxelles airport	US	84510
zaventem luchthaven	BE	84510
searchitall	CA	5856
search engine	US	5856

weight of the corresponding topic. Based on the click-through data, the similarity between any two queries can be calculated as the cosine similarity of the corresponding summing vector for each query. For example, query q is connected to $d1$, $d2$, and $d3$; query q' is connected to $d2$, $d4$, and $d5$, then the similarity between q and q' is the cosine between V and V', where

$$\vec{V} = \vec{d1} + \vec{d2} + \vec{d3} \; and \; \vec{V} = \vec{d2} + \vec{d4} + \vec{d5}$$

Suppose q comes from location $L1$, q' is from location $L2$, and we use q' to expand q. If $L1 = L2$, the expansion is called *same location-based query expansion*, else it is called *different location-based query expansion*. The *ground truth* is the list of ranked relevant document extracted from the search engine log data.

To formalize the observations, we define *location sensitive query* as follows:

Definition 1 (Location Sensitive Query) *: Given a query q, suppose the default search result list is Lf, the result list after same location based query expansion is Ls, the result list after different location based query expansion is Ld, and the ground-truth of the results list is L. Query q is defined as a location sensitive query if: $Q(Ls, L) > Q(Lf, L)$ or $Q(Ld, L) > Q(Lf, L)$, where $Q(Ls, L)$ is the quality of the returned results Ls compared against the ground-truth result L.* □

Furthermore, if $Q(Ls, L) > Q(Ld, L)$, then q is defined as a *same location sensitive query*; if $Q(Ld, L) > Q(Ls, L)$ then q is defined as a *different location sensitive query*.

The location difference in our experiments is identified at the country level. The quality of the returned results can be measured by different metrics such as Precision, MAP, NDCG, and Tau [32] [33].

To identify the location sensitive queries, we define another concept of *location sensitivity score*:

Definition 2 (Location Sensitivity Score) *: Given a query, q, and a list of relevant documents $\{d1, d2, d3, ...dn\}$ as the ground-truth of q. Suppose q is issued from m countries and the set of documents clicked by users from country i is represented as Di, then the location sensitivity score for query q is defined as:*

$$LSS(q) = \sum_{i=1, i \neq j}^{m} Sim(\sum_{ds \in Di} \vec{ds}, \sum_{dt \in Dj} \vec{dt})$$

where $0 < s, t \leq n$ and $0 < i, j \leq m$. □

The location sensitivity score describes the topic distribution of one query across different countries. A query is not location-sensitive or the location sensitivity score is 1, if users across different countries access the relevant documents with the same pattern. Here access pattern refers to the number of times a document was accessed in log data. The location sensitivity score is between 0 and 1. The larger the location sensitivity score, the less location sensitive the corresponding query.

Figure 3.2 shows the process of identifying the location-sensitive queries and classifying them into different groups for different location-based query expansion

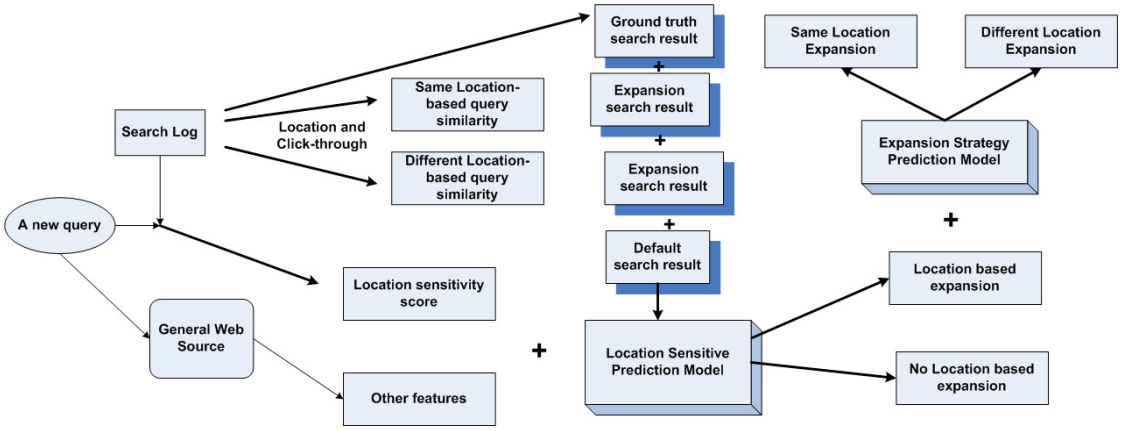


Figure 3.2. Identify location sensitive queries for different expansion strategies

strategies. From the search log, we first derive a list of queries to be expanded. Then the query expansion strategies of same location-based and different location-based are applied. The result quality of different types of query expansion is evaluated and applied to train a two-level SVM classification model. Once the model is constructed, when a new queries comes, we first extract its corresponding features and then predict its location sensitivity using the two-level classification model.

3.3 Two-level Query Classification Model

In this section, we explain how to create the two-level query classification model. At the first level, the queries are classified as location-sensitive or non-location-sensitive. At the second level, the location-sensitive queries are further grouped into same-location-sensitive or different-location-sensitive. To construct the classification model, there are three subtasks: feature extraction, label generation, and model training.

3.3.1 Feature Extraction

By mining the log data and using LDA, we extract eight critical features for each query: LSS, NO, NA, NA', NDO, NDA, NDA' and DLD. LSS is the location sensitivity score defined above. NO is the number of terms in the original query. NA is the number of terms added to the original query after different location-based query expansion, while NA' is the number of terms added to the original query after same location-based query expansion. NDO is the number of retrieved documents before query expansion. NDA is the number of retrieved documents after different location-based query expansion, and NDA' is the number of retrieved documents after same location-based query expansion. DLD describes the location diversity of documents related to a query. It is defined as the proportion of the number of documents clicked by users issuing query q to the number of countries where users issuing the query q:

$$DLD|q = \frac{|clicked\ documents\ of\ q|}{|countries\ where\ q\ is\ issued|}$$

3.3.2 Label Generation

Two labels of queries are used in SVM modeling: sensitivity label SL and type label TL. SL shows whether the query is location-sensitive, and TL shows whether the query is different-location-sensitive or same-location-sensitive. To generate the labels, we make three query expansion trials: same location-based approach, different location-based approach, and ignoring location-based approach. In the ignoring location-based approach, we only consider the query similarity from topic distribution vectors but ignore the location information.

The labels are decided by comparing the NDCG (defined in section 3.4.1) [33] of the three query expansion approaches. Given a query q, SL is set "+1" if a higher NDCG is obtained after different-location or same-location-based approach, and "-1" if the ignoring-location-based strategy gets a higher NDCG. TL is set "+1" if the NDCG of the different location-based approach is higher than the same location-based approach, and "-1" otherwise.

3.3.3 Classification Model Training

In the location-based query classification, we employ Support Vector Machines (SVM) [34] to generate a two-stage prediction model. After extracting the eight query features and two labels from the log data, we apply them to SVM and a two-stage model is generated to predict which type of query expansion should be applied to a given query.

This two-stage model can classify the queries at two levels. Given a query, if it is predicted as non-location-sensitive in the first stage model, no query expansion will be applied. But if it is predicted as location-sensitive, the second-stage stage model will be used. According to the second prediction result, the different-location or same-location-based expansion will be applied to the query.

3.4 Location-Based Query Expansion

In this part, we illustrate the topic-based document clustering and the application of location sensitivity in location-based similarity measure.

In topic-based document clustering, Latent Dirichlet Allocation (LDA) [14] is applied to generate the topic distribution. As the input of LDA, the document

collection contains all the documents reviewed by users in the click-through data. For each query, there is a list of reviewed documents related to it. By processing all the documents with LDA, a topic distribution on all the documents is produced. Each document is associated with a topic vector which specifies the topic distribution of the document.

Based on the topic vector representation, the location-based similarity measure is presented as follow:

For a given query q, there is a list of relevant documents $<d1, d2, d3, ...dn>$ that have has clicked by end users, where each document di is represented as a vector of topic distribution $<t1, t2, t3, ...tm>$ generated by LDA. Here m is the number of topics and i is between 1 and n. Then, the topic vectors of all di are summed up to generate a representative vector of q. The similarity between any two queries can be calculated as the cosine similarity between their representative vectors.

By taking into account of the location, we can propose a location-based query similarity measure as follows:

$$Sim(q_1, q_2|L_1, L_2) = Cos(\sum \vec{d_i}, \sum \vec{d_j})$$
$$(d_i \textit{ is clicked in response to } q_1|L_1,$$
$$d_j \textit{ is clicked in response to } q_2|L_2).$$

This equation means that for query $q1$ from location $L1$ and $q2$ from $L2$, their location-based similarity is calculated as the cosine similarity between their representative topic vectors. The documents associated with each query must be clicked by users from the same location, i.e., documents associated with $q1$ must be clicked

in response to searching with keyword $q1$ issued from $L1$, and the same for $q2$ and $L2$. Suppose the whole set of locations is C, if $L1 = L2$, then $Sim(q1, q2|L1, L2)$ is the same location-based query similarity, if $L1 = C - L2$, then $Sim(q1, q2|L1, L2)$ is the different location-based query similarity.

Initially, for each query q, its location-based similarity with all relevant queries are calculated based on its location sensitivity determined by the classification model. For example, if q is same-location sensitive, then the queries compared with it should be all issued from the same location as q. After that, among all relevant queries, the top K similar query terms are selected and added to the original query as expansion. In our implementation, K takes the value 1.

3.5 Experimental Validation

3.5.1 Set-Up

In experiments, we use 2G raw search log data from CiteSeer[1] (130,825 queries from 55,947 unique IPs) and 129,830 queries in search log data from Excite[2]. In experiments on Citeseer data, 281,379 documents clicked in the user log are used. In experiments on Excite data, 180,150 webpages are used.

In evaluation, we test our method over short queries, which contain no more than three terms. For Citeseer, we randomly select 30% from all the short queries, which are 3,863 in all. For Excite, we randomly selecte 2,400 short queries as test data.

The location of queries is identified at the level of country. Each IP address

[1] www.citeseer.ist.psu.edu
[2] www.excite.com

in the user log is mapped to a country. For the documents relevant to queries, we associate each query with the documents clicked by the same user in a time period of a maximum of thirty minutes. The same user is identified by the same IP address.

We use the clicked documents in the user logs as the baseline. When ranking the clicked documents for a given query, we refer to the default documents ranking by the search engine. This will lead to a positive enhancement to the default Citeseer/Excite performance in the evaluation experiments, especially the metric of NDCG and Kendall's Tau.

Four widely accepted evaluation metrics are used to evaluate our query expansion methods. They are Normalized Discounted Cumulative Gain (NDCG), Precision at $K(P@K)$, Mean Average Precision (MAP), and Kendall's Tau. They evaluate the query expansion methods from different aspects:

NDCG: NDCG is good for evaluating ranked result where relevance is ranged (e.g., 1 10) instead of binary. It is a metric focusing on both the ranking of documents and the precision of the result list. For a query q, given the ground truth R_1 and the results R_2 of any other approach, the NDCG of R_2 is calculated as

$$NDCG_q = M_q \sum_{j=1}^{K} \frac{(2^{r(j)} - 1)}{log(1 + j)}.$$

Here, the results in R_2 are examined with an order of top to down. For each result document in R_2, $r(j)$ is the relevance label which shows the relevance of the document at position j to R_1, and M_q is the normalized constant which makes the NDCG of the perfect R_2 as 1. In our experiment, we make the relevance label set

as follows: 4 means the document has the highest relevance to the ground truth R_1 and 0 means the document is not relevant at all. K represents the number of documents examined in R_2, which is set to 20 in our experiments.

Precision: Given two lists of rank results R_1 and R_2, where R_1 is the ground truth, R_2 is the results of any other approach, and the $P@K$ for that approach is defined as the proportion of documents in both R_1 and R_2 to the documents retrieved in R_2. Here we make $K = 20$ and use $R(20)$ to represent the top 20 documents in result set R; so $P@K$ can be computed as

$$P@20 = \frac{|R_1(20) \cap R_2(20)|}{|R_2(20)|}.$$

Mean Average Precision: MAP emphasizes the position of relevant documents in a returned list. Given the ground truth R_1 and results list R_2 of any other approach, average precision is the average of precision computed after each relevant document in R_2 is examined. MAP is the mean of average precision of all the queries. The earlier the positions of returned relevant documents, the higher the MAP. Assume N is the number of all queries, $P(j)$ is the precision at position j, then MAP can be computed as follows:

$$MAP = \frac{1}{N} \sum_{i=1}^{N} \frac{\sum_{j=1}^{K} (P(j) \times relevance(j))}{|relevantdocuments|}$$

Kendall's Tau: It is used to measure the correspondence of the ranking in two document lists. The value of Kendall's Tau is 1, if the correspondence of two rankings is perfect, and it is -1 if otherwise. For a query q, given the ground truth list R_1, assume that n is the number of items in R_2, d_j is the item at position j in R_2, and P_j is the number of documents appearing after d_j in both R_1 and R_2,

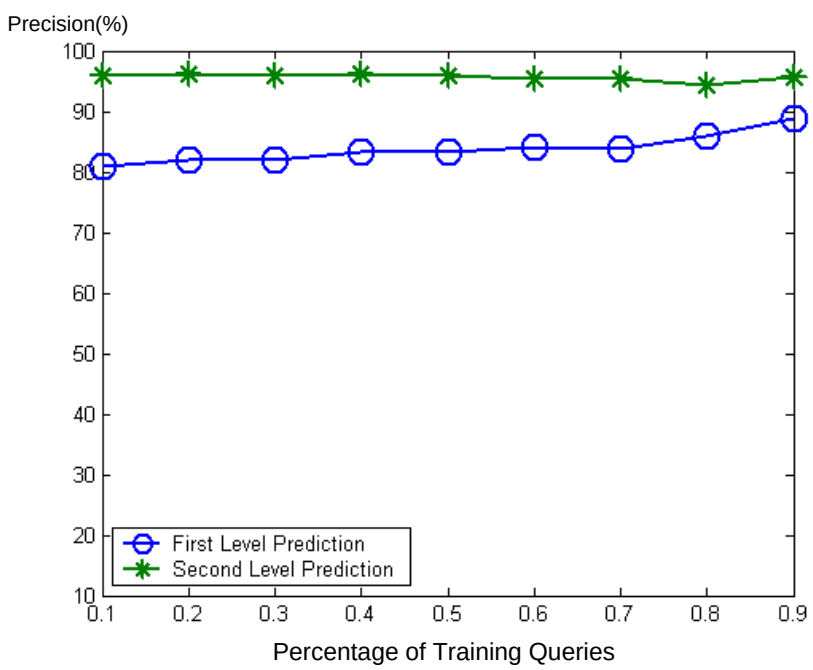


Figure 3.3. Precision of the two-level query classification model. The first level determines whether queries are location-sensitive. The second level determines the type of location sensitivity.

then the Kendall's Tau between R_1 and R_2 is calculated as follows:

$$\tau = \frac{2\sum_{j=1}^{n} P_j}{\frac{1}{2}n(n-1)} - 1$$

3.5.2 Evaluation on Citeseer Search Engine Log

In the experiments with Citeseer, we compare our location and topic-based method to the default Citeseer results, the corpus-based query expansion [35], and Wordnet based query expansion approaches [26].

First, to test the precision of the two-level query classification model, we conduct cross-validation on the 3,863 queries. Part of queries is randomly selected as training data, and the rest is used as test data. A group of experiments with an increased number of training queries and decreased number of predicting queries

Table 3.2. Comparison of four query expansion strategy on location sensitive queries with CiteSeer

Evaluation Strategy	Citeseer Default Strategy	Topic & Location-based Strategy	Corpus-based Strategy	Wordnet-based Strategy
Precision (%)	19.40	19.79	13.51	14.53
NDCG (%)	32.84	33.34	19.92	23.32
MAP(%)	85.60	92.24	54.72	62.36
Kendall's Tau	-0.923	0.001	-0.935	-0.930

are executed. Figure 3.3 shows the changes in predicting precision with an increase of the number of training queries. As Figure 3.3 shows, our two-stage SVM prediction model has good precision in predicting the query location sensitivity. The average precision of the first level and second level predictions are 83.78% and 95.76% respectively. In the first level, the maximal and minimal precision are 88.86% and 80.93%; while for the second level prediction, the maximum and minimum are 96.21% and 94.37%.

Second, with the ground truth and sensitivity labels, 702 location-sensitive queries are picked out from all the 3,863 queries. We compare our method with the default Citeseer result, the Wordnet-based query expansion result, and the Corpus-based query expansion result. The strategy of Wordnet-based query expansion method is to expand the initial query with synonyms picked out from the Wordnet. And the Corpus-based strategy is to calculate the relevance of words based on their context and then expand the initial query with the top relevant words.

Table 3.2 shows the Precision, NDCG, MAP, and Kendall's Tau of the four strategies on the 702 location sensitive queries. When comparing with default method, the topic and location-based method is 7.7% and 0.923 better in terms of MAP and Kendall's Tau. The improvements show that, even the ground truth has a positive effect on the default CiteSeer results, the document ranking of our

Table 3.3. Comparison of sub-strategies of location based method on 118 location sensitive queries with Excite

Evaluation Strategy	Excite Default Strategy	Topic & Different-location Strategy	Topic & Same-location Strategy	Topic & Ignoring-location Strategy
Precision(%)	2.67	3.40	3.40	3.14
NDCG(%)	9.46	20.76	14.25	20.05
MAP(%)	1.13	22.88	12.15	19.49
Kendall's Tau	-0.99	-1.0	-0.973	-1.0

method is much better than the default CiteSeer results. Also because the queries in Citeseer data all aim at academic documents, the effect of location on vocabulary diversity is reduced in some degree.

3.5.3 Evaluation on Excite Search Engine Log

In experiments with the Excite search engine, we compare the default Excite search result with three sub-strategies in our query expansion method. Among 2,400 short queries, 218 are location- sensitive, in which 123 queries are different location sensitive and 95 are same location-sensitive. In the results, the NDCG values of 1,563 queries with the four strategies are all 0. The probable reason is that the index of webpages on the Excite search engine changes over time. But with other queries, we can still see a significant improvement caused by our location and topic-based method.

For evaluation, we randomly select a sample of 1,000 short queries from the Excite log, in which 118 location-sensitive queries are detected. Table 3.3 shows the comparison of the four strategies on the 118 location sensitive queries. Because the precision is calculated at the base of 20($K = 20$) and the number of clicked webpage for a query is usually 1, the average precision values shown in Table 3.3 are close to 5%. On one hand, the boost of precision value shows that our method

has more precision; on the other hand, the significant increase of both NDCG and MAP values reflect the improvement of ranking caused by our method. Table 3.3 shows that our location and topic-based method produces significant improvement in general search engine log. Comparing the improvements on Citeseer data and Excite data, it is observed that the query location sensitivity is much more obvious in general webpages than in academic documents.

In summary, the experiments show that our topic and location-based method outperforms other query expansion approaches. On one hand, it can effectively select the location sensitive queries; on the other hand, for location sensitive queries, our query expansion methods significantly improve the search results. What is more, the experiments indicate that the location sensitivity of queries is not so strong in academic area. With the general search log from Excite, we can see that the location-based query expansion strategy performs much better.

Chapter 4

Mining Temporal Evolution of Social Networks

In social networks, the semantic relations embodied in UGC reflect the correlation between individuals. At the same time, the semantic topic distributions in UGC are also affected by the evolution of relevant communities. Therefore, investigating the patten of social network evolution helps us understand how the semantics in UGC change over time. In this chapter, we present an approach to study the member interaction on the active community evolution. A statistical model is adopted to measure the significance of different structural features on two types of social networks. Furthermore, we show by experiment that using the most significant structural features can improve the accuracy of predicting the community evolution.

4.1 Literature Survey

In recent years, the evolution of large scale social networks has been studied and explored from different perspectives. These studies involve social networks formed under different environments, such as world-wide web, blogger networks, online friendship networks, email communities, phone call network, academic co-authorship networks, etc. Typically, existing work can be categorized into three groups: static network mining [16], microscopic evolution prediction [17], and time-evolving structure analysis [36, 37].

Some structural properties are discovered by mining the snapshots of the static network. The power-law distribution is discovered to be a common feature when the scale-free network expands with cumulative new vertices [16]. Small-world is another phenomenon observed in [38]. A study on the web shows that its average diameter is small and the web forms a small-world network [39]. Furthermore, social networks were found different from other networks in two ways: that their network transitivity is nontrivial and that the degrees of adjacent vertices have positive correlations [40]. These studies revealed important properties, but they were performed only on the static graphs.

The prediction of social network evolution pays more attention to the attachment of new edges and the arrival of new vertices. The classic E-R model simulates the network growth when the edges between vertices are added randomly [41]. Different from that, preferential attachment of new vertices are proposed to capture the power-law degree distribution [17]. Subsequently, a generative model was proposed to accommodate both the scale-free distribution and patterns of the features, such as average distance and clustering coefficient [42]. Another micro-

scopic evolution model was developed with nodes arriving at a prespecified rate and selecting their lifetimes [43]. These models address the network evolution with a micro scope, but they did not consider the macroscopic structural predictors.

Time-evolving structure analysis focuses on the structural feature measures and their evolution over time [44]. The forest fire model is presented to explain the densification and shrinking diameters over time [45]. Furthermore, different behavior scaling in degree distribution is analyzed on various online social networks [46]. However, these studies only focus on the structure change and ignore the overall network evolution. Co-evolution of social and affiliation networks is addressed in [47]. Derived from the traditional two-step approach, an algorithm FacetNet is proposed for time-evolving community extraction and formation in [48]. By analyzing the co-authorship network and phone-call network, Palla et al. revealed some activity patterns of the members and the influence of their geographic locations [49]. Another analysis performed on the co-authorship network explored the properties of community growth as well as the topic change [50]. In these works, the influence of structural properties is considered at the individual level and their measures of growth are based on the cumulative additions of new vertices.

Previous studies either consider membership as permanent after the individual joins the network, or ignores the influence of structural properties when simulating the network evolution. Different from them, we focus on the active population within a social network and measure the impact of member interaction on the active community. At last, the results are used to improve the prediction accuracy of the community evolution.

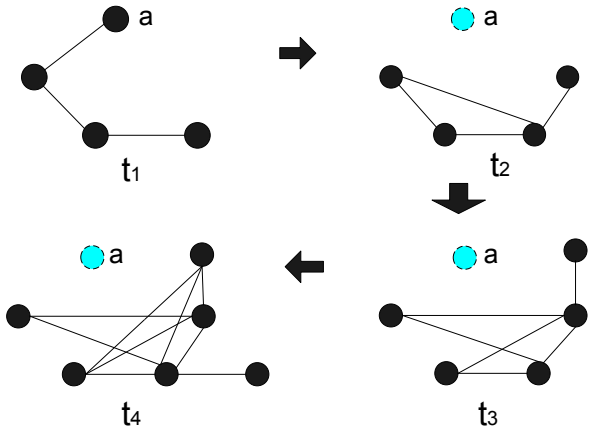


Figure 4.1. An example of Facebook social network evolution

4.2 Preliminaries

In social network analysis, existing models explore the social network evolution based on the cumulative membership. Once an individual joins a community, his/her membership is usually considered dormant no matter whether he/she has activity or not in the future. Based on this assumption and preferential attachment, the scale-free social network is observed to follow the power-law degree distribution. However, this assumption is not always consistent with the facts. An underlying intuition in evolution studies is that member interactions have effects on social network evolution and would determine its future status. If an individual does not have any activity during a time period, he/she may not contribute to the community evolution and thus should not be considered in the evolution study over that period.

Figure 4.1 shows an example of a community evolution in the Facebook wall-posting network over a period of four time steps. At time t_1, four nodes represent four members and they have totally four posts on the walls of each other. At time t_2, member a does not have any activity, but others keep active and a new member

joins in. At time t_3, more posts appear between other members and a second new member is attracted, while a is still inactive. At time t_4, the members are more active and the community keeps expanding, but a does not contribute into the activity nor the community growth.

From Figure 4.1, we can see that not all the members are consistently active after joining a community. The evolution of a social network tends to be affected by the activity between members. The more active the participants are, the more members the community is likely to attract. Additionally, not all members in a social network contribute to the growth of the community. New members are usually introduced into the community by the existing active members, instead of the inactive ones. Therefore the inactive members should not be included when evaluating the effects of member interaction on the community evolution. Based on these observations, we formalize the concept of *active social network* as follows:

Definition 3 (Active Social Network) *: Given a group of individuals M with interactive activities during a time interval (t_0, t_n), the active social network of M at time $t \in (t_0, t_n)$ is represented by $G(t) = (V_t, E_t)$, where V_t is the vertex set of the network and E_t is the edge set. Every vertex $v \in V_t$ corresponds to an individual who is involved in at least one activity at time t. An edge $e =< u, v >, e \in E_t$ exists between a pair of vertex u and v if and only if individual u and individual v have at least one interactive activity at time t. The vertex and edge sets, in the sense of their components, both vary according to time, i.e. $V_t = V(t), E_t = E(t)$.*$\square$

Within an active social network, the membership is no longer static or simply cumulative. Only those who have activities at time t are included in $G(t)$. Therefore, the membership and status of $G(t)$ evolves with time t. Since both the community topology and the membership are involved in the social network

evolution, it is hard to describe them at the same time. Focusing on the overall description of the social network, we propose to use the community size $N(t) = |V(t)|$ as the measure of network status. However, this measure could be flexible and is not restricted to $|V(t)|$. It may vary according to different applications. Based on the community measure, we define a new concept of *evolving pattern* to describe the evolution of active communities:

Definition 4 (Evolving Pattern) *: Using $y(t) = N(t)$ to scale the change of the community measure, the evolving pattern is $D = dy/dt$. In a discrete format, $D = \Delta y/\Delta t$, or $D(t) = (y(t+1)-y(t))/\Delta t$, where Δt is a fixed time interval. Based on different values of D, evolving pattern has two labels: growing and shrinking. Growing is when $D >= 0$, and shrinking is when $D < 0$* □

The evolving pattern describes the evolving status of an active community in successive time steps. In this study, we only focus on the labels of the evolving pattern, i.e., growing and shrinking, and use binary marks L to represent the two labels. The evolving pattern is marked as 1, if the social network is growing in the next time step, and 0 if shrinking.

The evolving pattern measures the evolution of a social network from the macro scope, instead of the micro scope. Taking no account of the random factors and the environment variables, it is the member interaction in $G(t)$ that is most likely to attract new members and determine the future community status. To study the influence of the member activities on the evolving pattern, it is necessary to understand the relationship between them. Although it is hard to describe the member activities directly, they could be summarized by various structural features of $G(t)$. In this way, the member activities can be described alternatively.

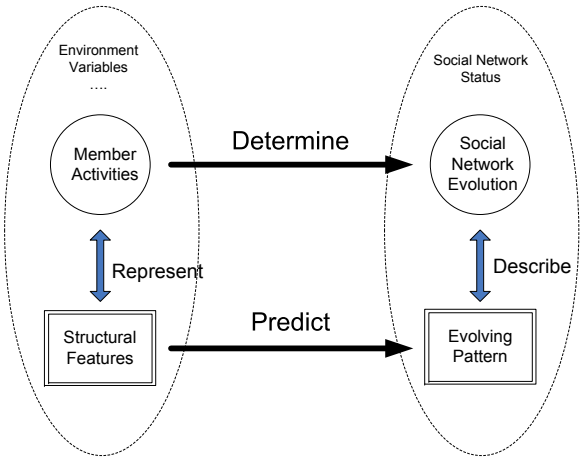


Figure 4.2. The relationship of the member activities and social network evolution

Figure 4.2 illustrates the interactive relationship between the social network evolution and the member activities as well as their alternative representation. The description of member activities are quantified by extracting and integrating different structural features. At the same time, the active social network evolution is measured by the temporal evolving patterns.

By using structural features, the problem of exploring the impact of the member interaction on the evolution can be transformed to finding the influence of structural features on the evolving pattern. Based on this, a quantitative model can be employed to measure their relationship, where the structure features are considered as the multi-dimensional predictor and the evolving pattern is the response. The model is expressed generally in the following format:

$$D \sim f(\{sf_i, i = 1, 2, ...k\}),$$

where D represents the evolving pattern, f represents the relation function, and sf_i represents the i-th of the total k structural features.

Once the function f is fitted, it can serve to explain the evolving pattern with the structural features. In this way, the influence of structural features is measured quantitatively and the evolving pattern can be predicted once the relation function is given.

In a social network, there are many structural features that have influence on the evolution. However, given the full feature set, not all of them have the same impact on the evolving pattern. Finding out those significant structural features is important to understand how the community evolution is determined by the member interaction. Additionally, it will also help reduce the complexity of the fitting model.

What is more, the full feature set may not produce the highest predicting accuracy. Better accuracies might be obtained with fewer features. One explanation could be that though more variables can reduce the prediction variance, the correlation between these variables may introduce much more bias into the result. This leads to the overall decrease of the predicting accuracy, called the "overfitting" problem. Therefore, it is necessary to select the most efficient set of structure features to predict the evolving pattern.

Based on the observations above, we have the following conclusions:

- Not all structural features are of the same significance in determining the evolving pattern.

- When measuring the impact on the evolving pattern, too many structural features may lead to the problem of overfitting.

Given the overall analysis, the principle of our research is to find out how the community evolution is affected by the member interactions and select as

Table 4.1. Structural features of CiteSeer Co-authorship Network

Notation	Structural Feature Description
N_t	The number of active members in social network $G(t)$
CN_t	The cumulative number of distinct members from $G(0)$ to $G(t)$
ΔN_t	The difference between N_t and N_{t-1}
P_t	The number of all publications at time t
CP_t	The cumulative number of all publication from time 0 to time t
ΔP	The difference between P_t and P_{t-1}
E_t	The number of edges in $G(t)$
CE_t	The cumulative number of edges from $G(0)$ to $G(t)$
ΔE_t	The difference between E_t and E_{t-1}
C_t	The number of all collaboration in $G(t)$
ΔC_t	The difference between C_t and C_{t-1}
CMC_t	The cumulative number of collaboration from $G(0)$ to $G(t)$
AR_t	The average number of collaborators of each person in $G(t)$
ΔAR_t	The difference between AR_t and AR_{t-1}
CC_t	Average clustering coefficient in $G(t)$
ΔCC_t	The difference between CC_t and CC_{t-1}
AL_t	Average length of the shortest pathes in $G(t)$
D_t	The diameter across all vertices of $G(t)$

few structural features as possible that produce the best predicting accuracy. In the next section, we introduce the structure features extracted from active social networks and apply a method to predict the evolving pattern. At the same time, another approach is adopted to select the most significant features in explaining and predicting the evolving pattern.

4.3 Structural Feature Extraction

Although many structural features are available to represent the member activities, it is necessary to include the most relevant ones into our pool. Let s_t denotes the status of a social network G at time t, then $P(s_t)$ is the probability of $G(t)$ to be in the status s_t. Considering that the member interaction at time t plays an

Table 4.2. Structural features of Facebook Online Wall-posting Social Network

Notation	Structural Feature Description
N_t	The number of active members in social network $G(t)$
CN_t	The cumulative number of distinct members from $G(0)$ to $G(t)$
ΔN_t	The difference between N_t and N_{t-1}
P_t	The number of all posts in $G(t)$
CP_t	The cumulative number of all posts from time 0 to time t
ΔP	The difference between P_t and P_{t-1}
E_t	The number of edges in $G(t)$
ΔE_t	The difference between E_t and E_{t-1}
CE_t	The cumulative number of edges from $G(0)$ to $G(t)$
AI_t	The average number of interaction per person in $G(t)$
ΔAI_t	The difference between AI_t and AI_{t-1}
AP_t	The average number of post for each person in $G(t)$
ΔAP_t	The difference between AP_t and AP_{t-1}
CC_t	Average clustering coefficient in $G(t)$
ΔCC_t	The difference between CC_t and CC_{t-1}
AL_t	Average length of the shortest pathes in $G(t)$
D_t	The diameter across all vertices of $G(t)$

important role in determining s_{t+1}, we assume that s_{t+1} is dependent on s_t and independent of all other previous status; then the probability of $G(t+1)$ to be in status s_{t+1} is as follows:

$$P(s_{t+1}|s_t, s_{t-1}, ...s_0) = P(s_{t+1}|s_t).$$

Given the assumption above, the evolving pattern marks L at time $t+1$ can be determined with the structural features of $G(t)$ and $G(t+1)$. At the same time, the structural features at time t depend only on s_t. Based on this inference, we extract the most frequently used structural features on each time step from both the CiteSeer and Facebook dataset. The feature sets include not only the characteristics related exclusively to the social network structure, but also those indicating the activity level of the members. Additionally, the features that serve

to explain the topology change of the social network are also generated, e.g. the average shortest path length. The radius and the diameter are computed based on the results from all the connected components. To measure the effects of the previous status, we also include some features that describe the community status change on two successive time steps.

Table 4.1 and table 4.2 summarize all the structural features generated from the CiteSeer and Facebook dataset. There are totally 18 structural features obtained from CiteSeer data and 17 from Facebook data. The structural features extracted for the two datasets are not exactly the same because of their different infrastructure. To handle the very large values of some features, they are rescaled by logarithms. These structural features are explored with a variable selecting procedure and the most significant ones are picked out. Furthermore, those selected features are applied to a fitting model and the evolving pattern prediction is performed.

4.4 Evolving Pattern Prediction

With all the structural features extracted, we adopt a shrinkage method to study the impact of member activities on community evolution. After that, we apply the shrinkage method with a statistical model to predict the evolving pattern with the most significant features.

4.4.1 The Shrinkage Method

Denoting the candidate structural features with variables, the problem of feature selection can be approached by shrinkage methods. When selecting variables, using

shrinkage methods allows a variable to be partly included in the fitting model. That is, the shrunken coefficient indicates how much information the variable contributes as a factor in the model. The Lasso is a widely accepted method and stands for "Least Absolute Shrinkage and Selection Operator" [51]. Utilizing the Lasso, we apply it with our fitting model and find out the most significant structural features.

Take Lasso with a linear fitting model as an illustration. Given a set of input variables $x_1, x_2, ...x_p$ and a response Y, the linear regression fitting model is expressed as:

$$\hat{Y} = \beta_0 + \sum_{i=1}^{p} \beta_i x_i.$$

The Lasso fits the model and estimates the coefficients $\beta = (\beta_0, \beta_1, ...\beta_p)$ by the criterion

$$\hat{\beta} = argmin(\sum (Y - \hat{Y})^2), \tag{4.1}$$

subject to

$$\sum |\beta_i| \leq s, \tag{4.2}$$

where $s > 0$ is a user-specified parameter.

The criterion of the Lasso is to minimize the residual sum of squares subject to the constraint (4.2), where the parameter s is often set moderately small. With the constraint (4.2), some of the solution coefficients can be shrunken to exactly zero. This makes the final model more interpretable. In application, the fitting model is not restricted to be linear regression. By using the Lasso, the coefficients β are estimated with (4.1) as well as (4.2), and the contribution of predictor variables is

reflected by them.

4.4.2 Prediction Model

As a shrinkage method, Lasso needs to be applied with an appropriate fitting model to measure the significance of the variables. Since the evolving pattern is labeled with binary marks, the response of the model is categorical, instead of numerical. Thus we adopt logistic regression as the fitting model, which has less assumptions on the decision boundaries and is more robust than the linear model. The results of logistic regression are probabilistic, instead of binary. They enable a flexible optimal boundary to assign the evolving pattern marks.

Suppose $Y_1, Y_2, ..., Y_n$ are independent binary response variables, which denote the evolving pattern marks and take the value of 1 or 0. The structural features are predictor variables represented by $x_1, x_2, ..., x_n$, with $x_i = (1, x_{i1}, x_{i2}, ..., x_{ip})'$, $1 \leq i \leq n$. p and n specify the number of variables and the size of the dataset respectively. Defining $\pi(t) = \frac{e^t}{1+e^t}$, according to the logistic regression, we have:

$$P(Y_i = 1|x_i) = \pi(x_i{}'\beta) = \frac{exp(x_i{}'\beta)}{1 + exp(x_i{}'\beta)}, \tag{4.3}$$

where $1 \leq i \leq n$, $\beta = (\beta_0, \beta_1, ...\beta_p)$ is a (p+1) dimensional vector of coefficients including the intercept.

According to (4.3), given an x_i, the logistic regression generates the probability of x_i taking the mark $Y_i = 1$. The optimal decision boundary of $P(Y_i|x_i)$ could be determined by achieving the best prediction accuracy on the training set. For simplicity, we take 0.5 as the decision boundary in this study. If $P(Y_i = 1|x_i) > 0.5$, then the response Y_i is 1 and the evolving pattern is labeled growing; else the value

of Y_i would be 0 with the label as shrinking.

4.5 Member Interaction Analysis

Accepting the logistic regression as our fitting model, we apply Lasso to measure the impact of different structural features on the community evolution

Based on equation (4.3) , the log-likelihood function of logistic regression is:

$$\tilde{l}(\beta) = \sum_{i=1}^{n} \{Y_i log\pi(x_i{}'\beta) + (1 - Y_i) log[1 - \pi(x_i{}'\beta)]\}.$$

Then the negative log-likelihood function can be expressed as

$$l(\beta) = -\sum_{i=1}^{n} \{Y_i log\pi(x_i{}'\beta) + (1 - Y_i) log[1 - \pi(x_i{}'\beta)]\}. \tag{4.4}$$

Using Lasso, we estimate β with the criterion

$$\hat{\beta} = \arg\min_{\beta} l(\beta), \text{ subject to} \sum_{j=0}^{p} |\beta_j| \leq s. \tag{4.5}$$

With the function

$$L(\beta, \lambda) = l(\beta) + \lambda \sum_{j=0}^{p} |\beta_j|,$$

the criterion (4.5) becomes equivalent to

$$\hat{\beta} = \arg\min_{\beta} L(\beta, \lambda), \tag{4.6}$$

where λ is a penalty parameter.

Using optimization techniques, the parameter β can be estimated with the

criterion (4.6). The value of β is dependent on λ. Once β is generated, the accuracy of the model can be obtained according to (4.3). In our computation, β is initialized to 0. By applying the maximum likelihood estimation with (4.6), β is updated and the prediction accuracy is calculated. Having the coefficient β, then among all the predictor variables, those having coefficients close to zero or significantly smaller than others will be removed. After that, a new β is calculated again with the maximum likelihood estimation, and then more variables are removed based on the new value of β. These steps are executed iteratively until no more predictor variable can be removed.

The procedure of the iterative algorithm for the significant feature selection is summarized as follow:

1. Fix λ and initialize the predictor variable set.
2. With the current variables, compute β that minimizes (4.6), i.e., the constrained maximum likelihood estimator.
3. Compute the accuracy based on β and current predictor variables.
4. According to β, remove variables with coefficients close to 0 or significantly smaller than the others.
5. Repeat steps 2, 3, and 4 until no more predictor variable can be removed.

Applying this algorithm, the significance of different structural features can be measured. The most significant feature set will be selected based on the best accuracy and the least predictor variables. Even with the same prediction accuracy, by choosing the more condensed variable set, the overfitting effect could be reduced more.

Based on the algorithm above, the impact of member interaction on community evolution can be determined. Once the most significant structural features are found out, they will be applied to predict the future evolving pattern. Therefore, active community evolution can be forecast given the current status. Furthermore, the semantic topic change in UGC can be explored together with the relevant community evolution.

4.6 Experimental Validation

In the experiment, we analyze the active community evolution on two types of social networks. Meanwhile, we evaluate the impact of member interaction on the evolving pattern and find out the most significant structural features. Finally, the experimental results show that certain structural features have significant impact on community evolution, and the accuracy of evolving pattern prediction can be improved with the selected features.

4.6.1 Set-Up

We measure the evolving pattern and evaluate the Lasso on both Facebook [52] friendship network and the CiteSeer [53] co-authorship network. In the experiment, to handle the very large values of some features, the first 12 features in CiteSeer data and first 9 features in Facebook data are rescaled by logarithm. The Facebook social network includes a total of 876,993 post records between 46,952 people over 1,596 days. Each record includes two anonymous user IDs and a time stamp, indicating that the second person post on the wall of the first person at the specified time. By setting the time step as one week and removing the records without any

collaboration, we generate the statistics of Facebook over 219 successive weeks. The structural features and the evolving pattern are extracted over each week. The maximum number of posts is 18,408 among 11,245 people in the 222^{ed} week. Among all the evolving pattern labels in the dataset, there are 141 growing and 78 shrinking, which are marked as 1 and 0, respectively.

The CiteSeer co-authorship network is collected from 1980 to 2006, which include 486,324 collaboration records and 283,155 authors. Different from the Facebook dataset, the structural features and evolving patterns of CiteSeer are measured annually. The maximum number of publications is 120,361 among 91,722 authors in year 2000. Also, the records with growing pattern are marked 1 and those with shrinking pattern are 0.

We use accuracy as the evaluation metrics. Let P denote the set of records with real growing pattern and Q denote that with real shrinking pattern. Assume that M is the set of records predicted to be growing and N are those predicted to be shrinking, then accuracy is measured as

$$Accuracy = \frac{|M \cap P| + |N \cap Q|}{|P \cup Q|}.$$

In computation of the feature selection, only one parameter λ is involved. To find an appropriate setting value, we calculate the coefficients of the model with different λ values. When λ is changed from 0.1 to 5, the fitted coefficients are the same. Based on that, we set λ as 1 in the computation.

For comparison, we use the decision tree as the baseline approach. The Gini index [54] is used as the impurity function when creating splits in the tree construction. For both the Facebook and the CiteSeer datasets, the structural features

are applied to the decision tree, and the most informative ones are selected as the attribute of splits at each level. The best pruned tree is determined according to the predicting accuracy with 5-fold cross-validation. After that, the attributes at splits in the best pruned tree are selected as the most important structural features.

The performances of the Lasso and the decision tree are compared based on the structural features selected by them. Additionally, the predicting accuracy of the logistic regression is compared against the best pruned decision tree.

4.6.2 Evaluation on Facebook Social Network

Among all data points of the Facebook dataset, we randomly select 25% data points for testing data and use the remaining 75% for training. We follow the iterative algorithm until it converges or no more structural features can be removed. In each iteration, the Lasso method is applied with the logistic regression to generate a predicting model on the training data, and then the predicting accuracy is calculated over the testing data.

Figure 4.3 shows the accuracy change on different numbers of structural features selected by the Lasso in each iteration. Error bars represent 5% errors. The method produces the best accuracy of 79.3% when only 2 features are included as the predictor. It is 5.7% higher than the accuracy obtained with a full feature set. Other feature sets are not considered as good choices, because they produce lower accuracy with higher complexity. Therefore, the set of 2 features is selected as the final result.

Although the full structural feature set includes 19 features, it does not produce high accuracy. One explanation could be that the correlation between structural

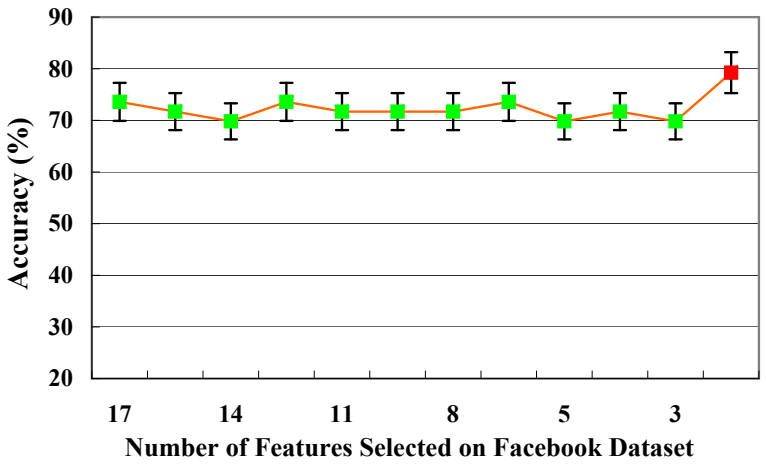


Figure 4.3. The accuracy over number of features on Facebook data by Lasso

Table 4.3. The comparison of structural features selected and accuracy between the Lasso and decision tree on Facebook data

Method	Lasso	Decision Tree
Structural Features Selected	N_t CE_t	D_t, P_t, ΔP_t ΔN_t,N_t AI_t,ΔAI_t, CC_t, ΔCC_t AP_t, ΔAP_t, ΔE_t
Accuracy	79.3%	69.7%

features introduces bias in the predictor. The more correlated features incorporated, the larger bias the predictor may have.

As the baseline, the decision tree is applied to the same training and testing data. Table 4.3 illustrates the comparison of our method and the decision tree in terms of predicting accuracy and the structural features selected by them. The predicting accuracy of our model is 9.6% higher than that of the decision tree. We can see that our method is effective in finding the most informative structural features. On the other hand, we observe that the selected features of both methods have N_t in common, and both results include a feature derived from the number of edges. It indicates that the number of current active members and features relevant to the number of edges could be important factors to explain the evolving

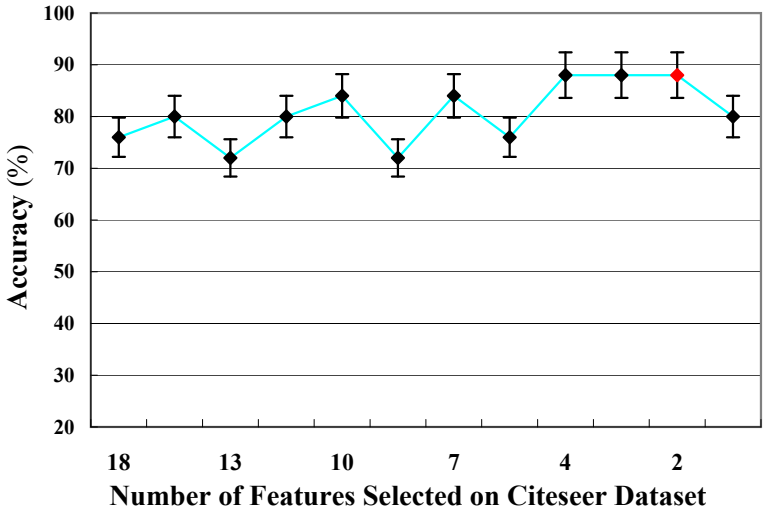


Figure 4.4. The accuracy over number of features on CiteSeer data by Lasso

Table 4.4. The comparison of structural features selected and accuracy between the Lasso and decision tree on CiteSeer data

Method	Lasso	Decision Tree
Structural Features Selected	AR_t C_t	ΔCC_t CN_t
Accuracy	88%	76%

pattern of online social networks.

4.6.3 Evaluation On Citation Social Network

In the second part of the experiment, our method is compared with the decision tree on the CiteSeer co-authorship network. The Lasso-based algorithm is applied to the CiteSeer data with the same iterative procedure. The predicting accuracy on different structural feature sets in each iteration is shown in Figure 4.4. Error bars represent 5% errors. The best accuracy 88% is obtained when the number of features used is 2, 3, or 4. According to our variable selection strategy, among them, the set of 2 features requires the least predictor variables when producing the best accuracy. Therefore it is selected as the final result. The 2 features selected

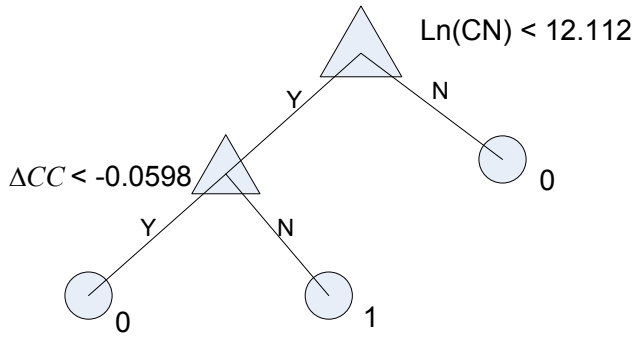


Figure 4.5. The decision tree grown on CiteSeer data

as the final result are shown in table 4.4.

For comparison, the decision tree method is also applied to the CiteSeer data. As a result, 2 structural features are selected as the split nodes of the best pruned tree, which is displayed in figure 4.5. The triangles represent the split nodes and the circles indicate the leaf nodes with evolving pattern marks. Table 4.4 shows the comparison of the structural features selected by the two methods and their corresponding predicting accuracy. Compared with decision tree, the features selected by the Lasso produce much better accuracy.

The structural features selected by the Lasso represent the total number of collaborations and average number of collaborators for each person, which are very close. This result reveals that the collaboration between coauthors could be the most significant characteristic to determine the evolving pattern of the co-authorship network.

In summary, the experimental results show that the member interaction have significant influence on active community growth and shrinkage. Applying Lasso and logistic regression, we find the most important structural features that can effectively determine community evolution. On different types of social networks, the

significant structural features found out are different. The evolution of friendship networks tends to be determined by the current active member and the cumulative correlations between individuals. In the co-authorship network, the number of collaborations between authors is the most significant feature to predict community evolution.

Chapter 5

Mining Sentiments and Topics from Social Media

In this chapter, we study sentiments and topics of UGC from social media furthermore. By observing topics and sentiments of online posts, we find that topics and sentiments are not completely independent. To investigate the correlation between, we propose a multi-task multi-label classification model that can classify sentiments and topics of tweets simultaneously. Experiments show that the proposed model has a higher classification accuracy than state-of-art methods.

5.1 Literature Survey

5.1.1 Multi-Label Classification

Multi-label classification is concerned with categorizing instances into multiple classes, while the associated classes are not exclusive. Each associated class of an instance is called a "label". Existing multi-label classification methods can be gen-

erally grouped into two categories: class reorganization and algorithm innovation.

Class reorganization methods reorganize classes to transform the multi-label classification into single-label classification. Three approaches are proposed for this purpose in [55]. They include: randomly selecting one from the multiple labels, ignoring all multi-label instances, and constructing multiple single-label classifiers. Another approach extends classes by constructing a label power set (LP) and considering each different label combination as a new class [56]. The disadvantages of this approach are that it may lead to a large number of reorganized classes and each class has too few instances. Another widely used reorganizing method is to construct a binary classifier for each class, and then the classification results on all classes are combined into a multi-label result [57]. In a methodology overview [58], an undocumented method is introduced, which decomposes instances by using only single labels and then merges the single-label classification results.

Algorithm innovation methods focus on modifying single-label classification models to adapt to multi-label classification. In [59], a mixture model is used to represent the multiple classes with training documents labeled by EM. An algorithm innovation with decision tree algorithm C4.5 adopts a new entropy measure that allows multiple labels in leaves [60]. After that, an algorithm MMAC is proposed, which learns a set of association rules first and then combine these rules into a multi-label classification model [61]. Jin et al. study a special kind of classification in which each instance is given a set of candidate labels and only one of them is correct [62]. In this work, a log-likelihood based approach is used together with EM to handle the multiple-label. Most existing multi-label classification methods cannot be directly applied to address multi-task classification. At the same time, the association between different tasks are not explored either.

5.1.2 Multi-Task Classification

Multi-task classification utilizes the correlation between related tasks to improve classification by learning tasks in parallel. Existing work mostly falls into two groups. The first group uses kernels and regularizers, while the second group investigates common features and task similarity measures.

Many algorithms are proposed to solve multi-task learning with various kernels and regularizer. In [63], k-nearest neighbor and kernel regression are introduced to learn tasks in parallel. Evgeniou et al. present a multi-task learning approach based on the minimization of a regularization function similar to the one of SVM [64]. Later, a multi-task kernel function is derived to help estimate multiple task functions at one time [65]. In [66], a multi-task learning algorithm based on gradient boosted decision tree is proposed for web-search ranking over multiple datasets.

Exploring common features and task similarity also helps with multi-task learning. Ben-David et al. define and exploit task relatedness by the similarity between distributions generated by examples of tasks [67]. Later, a common feature selection method is derived for SVM when multiple tasks exist over a common input space [68]. To learn some common features across multiple related tasks, a 1-norm regularization method with a new regularizer is introduced in [69]. In [70], a dirichlet process based model is proposed to identify similar tasks and solve both symmetric and asymmetric multi-task learning. Another study of features uses hashing to reduce feature dimension and apply it on very large scale multi-task learning.

These methods focus on multi-task classification but do not consider multiple labels in each task. The study of multi-label multi-task learning still remains open.

5.1.3 Tweet Sentiment and Topic Analysis

Tweet sentiment and topic analysis becomes very popular recently. However most state-of-the-art studies address only sentiment classification or topic classification. To determines tweet sentiment, query-based dependent features and related tweets are explored and incorporated in [20]. In [22], POS-specific prior polarity features are introduced and applied with a tree kernel for sentiment analysis. Tan et al. find that including the influence of social connections can improve accuracy of sentiment classification [21]. In addition, a graph model is introduced to classify sentiment of hashtags in a time period [19].

To classify topics of noun phrases in tweets, a community-based method is presented to identify their boundaries within the context and classify them to a specific category [71]. After that, a model that switches between two probability estimates of words is proposed, which can learn from stationary words and also respond to bursty words [23]. In [72], another method is introduced to determine whether a tweet is related to a topic or not by using data compression. Furthermore, a Bag-of-Words approach and a network-based approach are evaluated in classifying twitter trending topics into 18 general categories [73].

These approaches focus on single-label classification on either sentiment or topic classes. Among the state-of-the-art work, none of them studies multi-label classification that analyzes both sentiments and topics at the same time. To address the problem of multi-label multi-task classification, we propose an algorithm based on multi-label learning and utilize association between tasks to promote classification accuracy.

5.2 Preliminaries

Sentiment and topic analysis of social media have a wide application in business marketing and customer care. For instance, when promoting a new policy or a product, the company wants to know how customers comment about it so that they can respond properly and timely to address criticisms and issues. For this purpose, monitoring the current sentiment trend and topics towards a certain product or brand name is both necessary and important. However, as a lot of posts may be generated in a short time, hiring human experts to work on them is too expensive. To address this problem, it requires some techniques that can classify tweet topics and sentiments automatically and quickly.

However, sentiment and topic analysis of social media involves a lot of challenges. As tweets are very short and may contain incomplete sentences, their meaning could be ambiguous and interpretations highly rely on the context. At the same time, people tend to use informal language or even bad syntax in tweets. This makes classic methods of natural language processing not well applicable in many situations. What is more, topic classification is hard even if done by human experts. On one hand, topics of tweets may not be perfectly exclusive. On the other hand, the content of a tweet may cover multiple topics. Therefore, binary classification may not produce satisfactory results. To solve this problem, multi-label classification is required.

As we have introduced, tweet topics and sentiments are not completely independent. By observing a collection of tweets, we find that certain association exists between tweet topics and sentiments. In addition, the appearance of some terms may also serve as strong indicators of certain classes. As an example, Table 5.1

Table 5.1. Example Tweets of "Virgin Mobile" with Sentiments and Topics

ID	Content	Sentiments	Topics
1	Virgin Mobile's #Sparah campaign is genius! Love the episodes!	Positive	Compliment
2	I love the new phone u came out with for virgin mobile. i love the samsung restore.	Positive	Compliment
3	@virginmobileus Care to answer???	Negative	Complaint, Care/Support
4	is seriously annoyed with Virgin Mobile. Get your crap together and fix my account!!!!!	Negative	Complaint, Care/Support
5	@anonymizedName get the hell out of here with virgin mobile crap!	Negative	Complaint

shows some real tweets regarding "Virgin Mobile", with user names anonymized. Tweet sentiments are positive, negative, and neutral. Tweet topics are 10 predefined classes. As shown in the table, tweets 1 and 2 indicate an association between Positive sentiment and topic Compliment. These tweets both contain the term "love", which gives a strong indication for both Positive sentiment and topic Compliment. Tweets 3-5 are negative, while their topics include Complaint and Care/Support. They imply that these two topics are likely to appear together with Negative sentiment. Meanwhile, the term "crap" appears in both tweets 4 and 5, implying an association with Negative sentiment and those two topics.

As observed above, sentiment classification and topic classification are associated. What is more, these two tasks are also connected with certain indicating terms. Considering the association between tasks, co-classification of multiple tasks can help reinforce each other and produce better results than doing them independently. Meanwhile, each task may involve multiple labels, i.e. a tweet refers to more than one topic. Classifying with multi-label can help handle the class ambiguity and improve classification accuracy. Therefore, we propose to incorporate multiple labels into multi-task classification. In this way, we can make good use of the latent information in predicting features, and at the same time, employ the results of multiple tasks to promote each other.

To incorporate both multi-task and multi-label classification, we investigate the

following questions: *how to make use of multi-task classification to promote each task? How to incorporate and process multiple labels in multi-task classification? In particular, how to apply the method on sentiment and topic classifications?* Formally, the multi-task multi-label (MTML) classification is defined as follows:

Problem 1 (MTML Classification). *Given an instance x and classification tasks $T = \{T_j : j = 1, ...M\}$, where the j-th classification task T_j has a finite set of classes $L_j = \{l_{jk} : k = 1, ...K_j\}$, the goal of MTML classification is to find a collection of class label sets $Y = \{Y_1, ..Y_j..\}$ that x belongs to, $Y_j = \{l_{j1}, ..l_{jq}\} \subseteq L_j$ is the set of class labels of x for the j-th classification task.*

5.3 Overview

By classifying both sentiments and topics at the same time, in the MTML model, we incorporate the results into predicting features, so that labels of the two tasks can promote and reinforce each other. For each task, the model is trained with maximum entropy on different predicting feature spaces. To learn with multiple labels, model coefficients are estimated with an optimization of multi-task likelihood and the prior label distributions.

Figure 5.1 illustrates an overview of the classification using the MTML model for classifying sentiment and topic of tweets. With a tweet collection, first, we extract sentiment and topic predicting features. Meanwhile, by using an existing classification method or Amazon Mechanical Turk based crowdsourcing, initial class labels can be obtained. Then, incorporating initial labels with predicting features, we get compound sentiment and topic features. The model can be trained by estimating coefficients with the training dataset. Once the model is trained, given

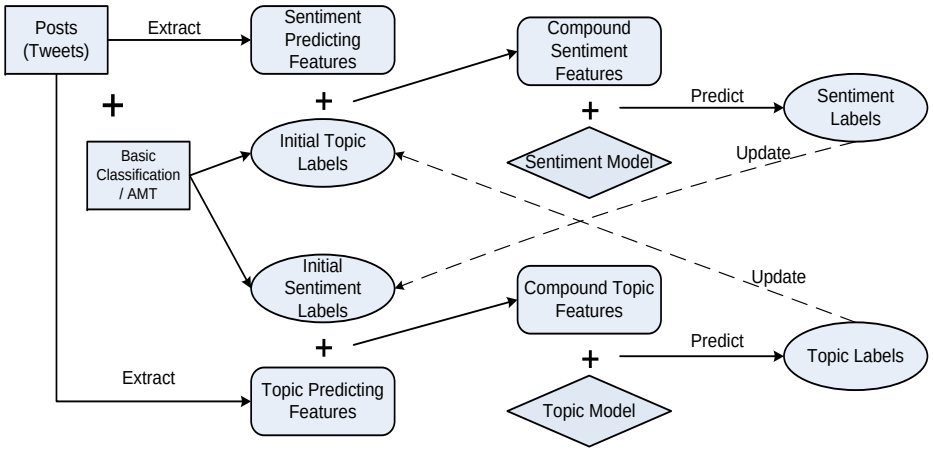


Figure 5.1. Multi-task multi-label classification model for both sentiment and topic classifications

compound features, it can generate new sentiment and topic labels. Repeating the two classifications iteratively can keep the class labels updated until it converges.

5.4 Multi-task Multi-label Classification Model

5.4.1 Feature Extraction and Selection

To train the MTML classification model, we first extract predicting features from tweets. Given a collection of tweets, we remove stopping words and select all keywords and bi-grams. For each tweet, its predicting feature vector $X_i = [a_1, a_2, ...a_m]$ consists of keywords and bi-grams in it. Since there are a tremendous amount of predicting features, feature selection is necessary to obtain the optimal predicting accuracy. For this purpose, using the Mallet [74], we measure the predicting accuracy of Support Vector Machine, Naive Bayes, and Maximum Entropy with different numbers of predicting features. Then we compare the results and determine the optimal number of predicting features accordingly. Feature extraction

and selection are conducted on both sentiment and topic classifications. The optimal predicting feature sets are selected separately on the two tasks. On different tasks, the number of optimal predicting features may vary.

5.4.2 The MTML Model

Within the predicting feature space, each tweet can be mapped to a feature vector. As we have introduced, each tweet instance is associated with a set of class labels. Assume that there are a total of K classes and N training instances. Let X_i denote the feature vector of the i-th instance x_i, where $i = 1, 2, ..., N$, and L_i denotes its label set. We apply Maximum Entropy (ME) to estimate the class distribution, which allows flexibility in model construction and also produces probabilistic classification result.

Let θ_k represent the coefficient vector of the k-th class, $k = 1, 2, ..., K$ and Y_i represent the class that instance x_i is assigned. Then, the probability of x_i to be classified into the k-th class becomes:

$$P(Y_i = k|X_i, \theta) = \frac{e^{\theta_k \cdot X_i}}{1 + \sum_{j=1}^{K} e^{\theta_j \cdot X_i}} \tag{5.1}$$

When solving the multi-task classification, we do not assume the independence of each task any more. By extending equation (5.1), we propose to incorporate classification labels of another task to make use of the latent task associations. Given an instance x_i, assume LS_i is its sentiment labels, and LT_i is its topic labels. Then, the feature vectors can be extended by including labels of another task. For the multi-task classification, let xs_i represent the sentiment feature vector and XS_i be the extended sentiment feature vector. Then, $XS_i = [xs_i, LT_i]$.

Similarly, use xt_i and XT_i to denote the initial and extended topic feature vector, $XT_i = [xt_i, LS_i]$. Based on them, let P_s and P_t denote the sentiment and topic distribution of an instance. Then, for the sentiment classification, we get:

$$P_s(Y_i = k|xs_i, LT_i, \theta s) = \frac{e^{\theta s_k \cdot XS_i}}{1 + \sum_{j=1}^{K} e^{\theta s_j \cdot XS_i}} \tag{5.2}$$

For the topic classification, next, we get:

$$P_t(Y_i = k|xt_i, LS_i, \theta t) = \frac{e^{\theta t_k \cdot XT_i}}{1 + \sum_{j=1}^{K} e^{\theta t_j \cdot XT_i}} \tag{5.3}$$

Now, we incorporate multi-labels into the classification. While learning with multi-label, our goal is to find the parameters θs and θt that maximize the probability of instance x_i to be labeled with LS_i and LT_i. Formally, let Θ denote the optimal values of $(\theta s, \theta t)$. Then, the objective function to estimate parameters can be written as:

$$\begin{aligned} \Theta = \arg\max_{\theta s, \theta t} \Pi_i P_s(Y_i \in LS_i|xs_i, LT_i, \theta s) \\ \cdot P_t(Y_i \in LT_i|xt_i, LS_i, \theta t) \end{aligned} \tag{5.4}$$

Let $\hat{P}_s$ and $\hat{P}_t$ be the prior probability generated from the labels. Then, P_s and P_t are the posterior probability produced by the classification model. To estimate parameters, one approach is to make the model based classification match the distribution from prior labels, i.e., minimize the difference between them. For each instance x_i, $\hat{P}_{s_i}$ can be calculated by the proportion of each label in LS_i out of all labels in LS_i; and similarly for $\hat{P}_{t_i}$. Both $\hat{P}_{s_i}$ and $\hat{P}_{t_i}$ are calculated with constraints of probabilities, $\sum_{k \in LS_i} \hat{P}_{s_i}(Y = k|x_i) = 1$, and $\sum_{k \in LT_i} \hat{P}_{t_i}(Y = k|x_i) = 1$.

Based on equation (5.4), a widely accepted method of parameter estimation is to minimize the KL-divergence between the prior and posterior probabilities of each instance. Denote S as all of the sentiment classes and T as all of the topic classes. Then, following the KL-divergence, the objective function can be furthermore written as:

$$\Theta = \arg\min_{\theta s, \theta t} \begin{cases} \sum_i \sum_{k \in S} \hat{P}_{s_i}(Y = k|x_i) log \frac{\hat{P}_{s_i}(Y=k|x_i)}{P_{s_i}(Y=k|xs_i, LT_i, \theta s)} \\ \\ \sum_i \sum_{k \in T} \hat{P}_{t_i}(Y = k|x_i) log \frac{\hat{P}_{t_i}(Y=k|x_i)}{P_{t_i}(Y=k|xt_i, LS_i, \theta t)} \end{cases} \tag{5.5}$$

Since for any class k that is not in LS or LT, the prior probability $\hat{P}_{s_i}(Y = k|x_i) = \hat{P}_{t_i}(Y = k|x_i) = 0$, having no influence on the parameter estimation. Therefore, equation (5.5) can be simplified to the following:

$$\Theta = \arg\max_{\theta s, \theta t} \begin{cases} \sum_i \sum_{k \in LS_i} & \hat{P}_{s_i}(Y = k|x_i) \\ & \cdot log P_{s_i}(Y = k|xs_i, LT_i, \theta s) \\ \sum_i \sum_{k \in LT_i} & \hat{P}_{t_i}(Y = k|x_i) \\ & \cdot log P_{t_i}(Y = k|xt_i, LS_i, \theta t) \end{cases} \tag{5.6}$$

with constraints $\sum_{k \in LS_i} \hat{P}_{s_i}(Y = k|x_i) = 1$, and
$\sum_{k \in LT_i} \hat{P}_{t_i}(Y = k|x_i) = 1$.

In equation (5.6), $\hat{P}_{s_i}$ and $\hat{P}_{t_i}$ are calculated from the labels. P_{s_i} and P_{t_i} are model-based probabilities, which vary with θs and θt. By solving equation (5.6), θs and θt can be determined. When the data is sparse, ME may have the problem of "overfitting." To reduce such an overfitting, we integrate the Gaussian prior into

ME for parameter estimation, with mean at 0 and variance of 1.

After the model is trained, given a tweet and the feature vector, its sentiment and topic classes can be determined by equation (5.2) and (5.3). Since extended feature vectors of the two tasks make use of labels from each other, it is necessary to obtain the initial labels. They can be generated from the classic ME model or any other classification approach. After that, during the process of multi-task classification, the sentiment labels obtained from equation (5.2) can be applied in equation (5.3) for topic classification, and vice versa. Repeating the two tasks iteratively will keep updating the classification results until it converges.

As a summary, the MTML classification proceeds as follows:

1. Given an instance x_i, extract its topic feature vector xt and sentiment feature vector xs.

2. Generate initial topic labels LT and sentiment labels LS of x_i by using a simple classification method or crowdsourcing.

3. Integrate LT with xs to obtain the compound sentiment feature vector XS, and obtain the compound topic feature vector XT similarly out of LS and xt.

4. Apply XS to the MTML *sentiment* classification model and generate sentiment labels LS' of x_i.

5. Apply XT to the MTML *topic* classification model to generate topic labels LT'.

6. Plug in LT' to update XS, and also use LS' to update XT.

7. Repeat steps 4-6 until the classification result converges.

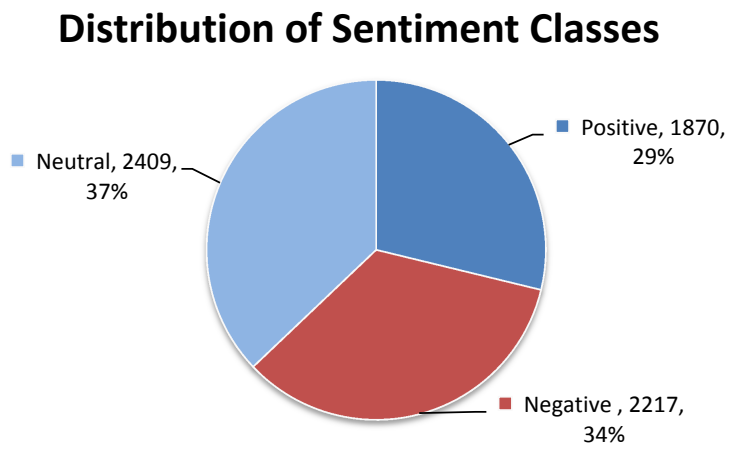


Figure 5.2. Percentages and numbers of tweets on sentiment classes

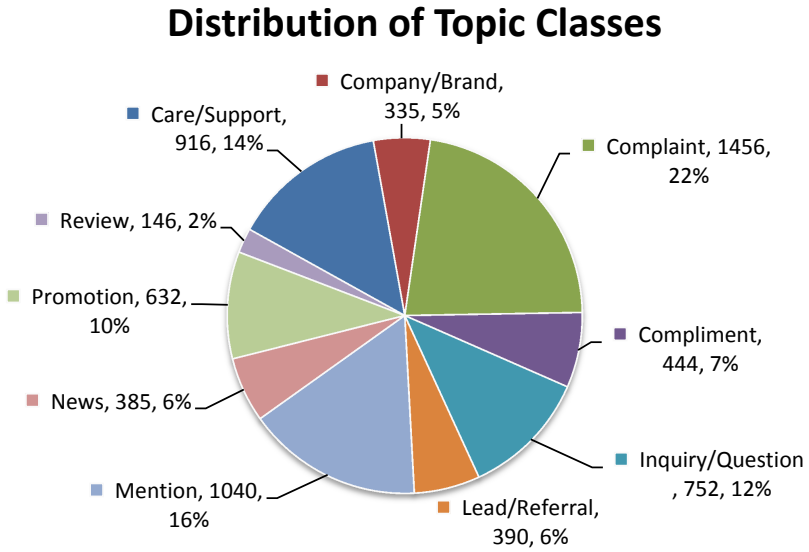


Figure 5.3. Percentages and numbers of tweets on topic classes

5.5 Experimental Validation

5.5.1 Set-Up

Dataset. The proposed MTML model is evaluated using real tweets crawled from 8/31/2010 to 4/26/2011. They contain at least one of the keywords "virgin-mobile", "VMUcare", "boostmobile", and "boostcare." Our target is to classify

sentiments and topics of these tweets towards "boost mobile" and "virgin mobile". After removing tweets that are posted by company customer services, we get a total of 6,496 user-generated tweets for the experiment. For classification, we take 3 sentiment classes and 10 topic classes, which are preset by professionals from the agent of the company "Virgin Mobile." The sentiment classes are "Positive", "Negative", and "Neutral". Figure 5.2 shows the number of tweets in each sentiment class and their percentage in the distribution. Topic classes include "Care/Support", "Lead/Referral", "Mention", "Promotion", "Review", "Complaint", "Inquiry/ Question", "Compliment", "News", and "Company/Brand". The number of tweets in each class and their percentages in the distribution are shown in Figure 5.3.

Ground-Truth. Initial sentiment labels and topic labels of tweets are assigned by crowdsourcing via Amazon Mechanical Turk (AMT). AMT is a crowdsourcing marketplace which allows collaboration of people to complete tasks that are hard for computers to do but easy for human workers to do. AMT has two types of users: requesters and workers. Requesters post Human Intelligence Tasks (HITs) with monetary incentives, while workers can browse HITs and complete them for monetary incentives. Requesters may accept or reject the result submitted by workers. With certain quality control mechanisms (e.g., majority voting or controlled HIT) requesters can obtain high-quality results for the submitted HITs through AMT.

Using the AMT, we collect 3 sentiment labels and 3 topic labels for each tweet. Labels may be identical or different. For each tweet, if at least two labels agree with each other, then this label is selected as the majority-voted label. Out of all 6,496 tweets, 6,143 of them have majority-voted sentiment labels, and 4,466

of them have majority-voted topic labels. Among 4,257 tweets that have both sentiment and topic majority-voted labels, we randomly select 500 for testing. The remaining ones and all other tweets that have 3 different labels are used as training instances, which contain 5,996 tweets. Since our MTML model can train with multiple labels, we make use of all labels in training. For testing, the majority-voted label is employed as the ground truth.

Baseline. To validate our model, we use 2 class reorganization methods: Label Power set (LP) [56] and Decompose-Merge Instance (DMI) [58], as well as 4 existing classification models as baselines. They include Naive Bayes (NB), Maximum Entropy (ME), Support Vector Machine (SVM), EM with Prior on Maximum Entropy (EPME) [62]. First, the MTML model is compared against the baseline models on both tasks. After that, we apply LP with DMI to convert the multi-task multi-label classification into single-task single-label classification first, and then measure the performance of baselines accordingly.

Feature Selection. Predicting features are first generated by extracting keywords from tweet contents. Hashtags are treated the same as other keywords, without any special weighting or discrimination. Initially, 50,553 keywords (thus feature dimensions) are extracted. Instead of doing dynamic feature reduction using conventional methods such as PCA, we used a simple empirical approach. We first measured the accuracy while varying the number of features from 400 to 5,000. For the sentiment classification task, the highest accuracy was obtained with 3,400 features, while for the topic classification task, 2800 features produce the best result. As a result, in the experiment, we simply adopted the 3,400 and 2,800 features for both sentiment and topic classification tasks, respectively. Note

that these two sets of features are independent. They are not combined together in the evaluations of our model and baselines.

Evaluation Metrics. We use classification accuracy to measure the performance of model. It is defined as follows:

$$Accuracy_{classification} = \frac{1}{N}\sum_{i=1}^{N} I(Z_i = Y_i)$$

where $I(true) = 1$ and $I(false) = 0$.

5.5.2 Evaluation on Twitter Steam Dataset

In the experiment, we evaluate MTML on both sentiment and topic classification tasks. The results of MTML are compared against baselines respectively. After that, we measure the average accuracy of MTML on multi-task and compare it against baseline results on the LP-converted dataset. In particular, we look into the classification accuracy on each class. By associating the class distribution with the accuracy improvement, we analyze their correlation and how the class properties affect accuracy.

First, we measure the MTML model on sentiment classification. The training dataset contains 5,996 tweets and the testing data contains 500 tweets. Each training tweet is associated with 3 training labels. Meanwhile, MTML is evaluated against NB, ME, SVM, and EPME. Figure 5.4 shows the accuracy of MTML and baselines on sentiment classification. As shown in the figure, MTML outperforms all baselines, achieving the accuracy of 0.744. Compared to ME and EPME, MTML makes an improvement of 5%. Although sentiment classification

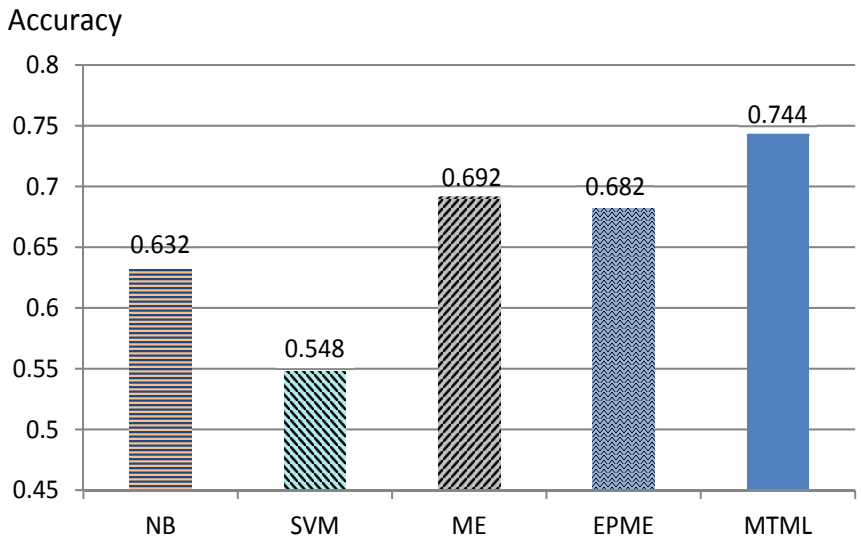


Figure 5.4. The accuracy of sentiment classification of five methods

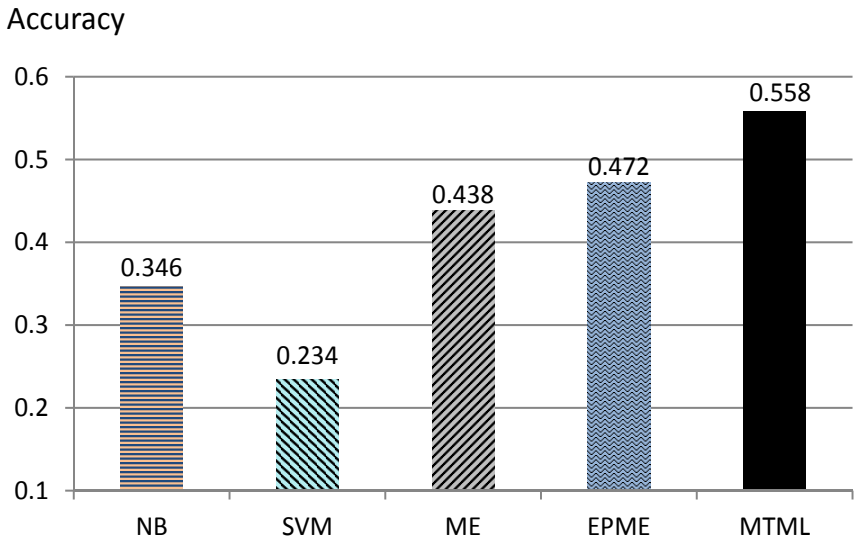


Figure 5.5. The accuracy of topic classification of five methods

achieves a fairly good accuracy with baselines already, therefore, using multi-task and multi-label enables a reasonable improvement.

Second, our MTML model is validated with topic classification on the same dataset. Classification accuracies of our model and baselines are shown in Fig-

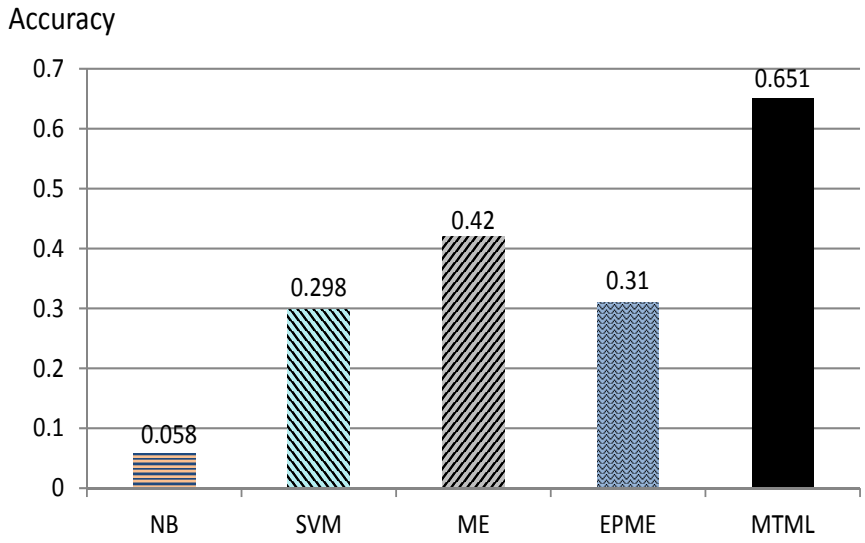


Figure 5.6. The accuracy of multi-task classification of five methods *after* class reorganization is applied

ure 5.5. Since there are a total of 10 topic classes and their distribution is not even, the accuracies of both MTML and baselines are not very high. However, MTML still outperforms the baselines and achieves an accuracy of 0.558.

Next, we use LP to transform the dataset into single-task classification with 30 classes (i.e., 3 sentiments $\times$ 10 topics). Furthermore, for each instance with multi-label, we apply DMI to convert it into multiple instances with single labels. Then, the accuracies of NB, ME, SVM, and EPME are measured on this converted dataset. Figure 5.6 shows the performance of MTML on multi-task classification against baselines after this class reorganization. Among all the methods performed, NB has the lowest accuracy while our proposed MTML still outperforms all baselines.

Since different classes take different proportions out of the whole dataset, next, we look into sentiment and topic classes and measure accuracy per each class. Figure 5.7 shows accuracies of all methods on each sentiment class. Overall, MTML

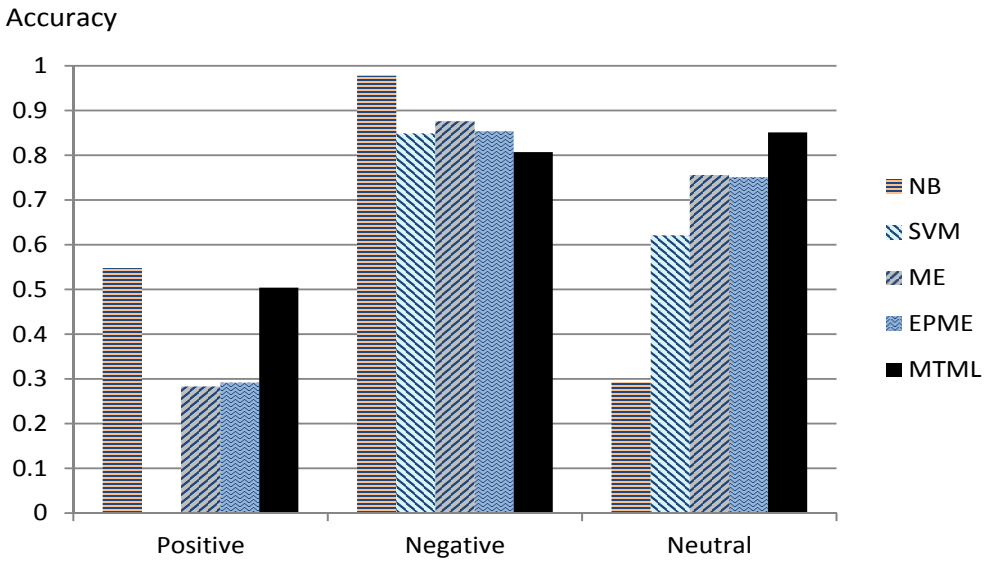


Figure 5.7. The accuracy of sentiment classification of the MTML model per three sentiment classes

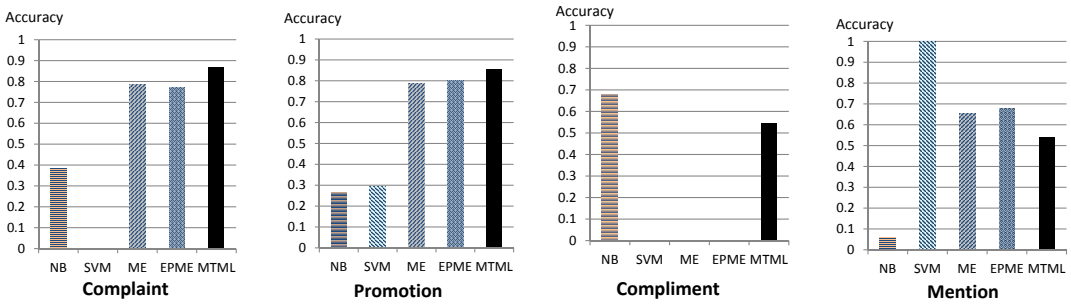


Figure 5.8. The accuracy of topic classification of the MTML model per four topic classes

performs well on all sentiment classes, especially the class of "Neutral". For topic classes, we show the comparison of all methods on 4 most interesting topics in Figure 5.8. Among all 10 topic classes, MTML tends to classify better on those relatively large-sized and explicit ones, such as "Complaint" and "Promotion." Figure 5.8(a) and (b) shows that MTML has an accuracy of 0.869 and 0.853 on these two classes, respectively. Another interesting observation is that NB outperforms the MTML model on small-sized classes, such as "Compliment" and "Review." As

Table 5.2. Sample tweets and topic classification results of NB, SVM, ME and MTML

ID	Tweet Content	Truth	NB	SVM	ME	MTML
1	Brought to you by boost mobile unlimited plan.....now with shrinkage????	Complaint	Compliment	Mention	Mention	Complaint
2	I am loving #Sparah and my @virginmobileus LG Optimus!!! @anonymized is so beautiful and ready for the spotlight.	Compliment	Compliment	Mention	Inquiry/ Questions	Compliment
3	New Boost Mobile Android phone for sale! The New Galaxy Prevail Touch Screen! If u want it get @me!	Promotion	Lead/ Referral	Mention	Mention	Promotion
4	That's top-up card It's a phrase which I believe was coined by virgin mobile for its prepaid phone service.	Mention	Compliment	Mention	Complaint	Mention

Table 5.3. Sample tweets and sentiment classification results of NB, SVM, ME and MTML

ID	Tweet Content	Truth	NB	SVM	ME	MTML
1	Brought to you by boost mobile unlimited plan.....now with shrinkage????	Negative	Positive	Neutral	Neutral	Negative
2	I am loving #Sparah and my @virginmobileus LG Optimus!!! @anonymized is so beautiful and ready for the spotlight.	Positive	Positive	Negative	Neutral	Positive
3	New Boost Mobile Android phone for sale! The New Galaxy Prevail Touch Screen! If u want it get @me!	Positive	Positive	Neutral	Neutral	Positive
4	That's top-up card It's a phrase which I believe was coined by virgin mobile for its prepaid phone service.	Neutral	Negative	Negative	Negative	Neutral

shown in Figure 5.8(c), NB is 13% better than MTML on "Compliment" class. Finally, Figure 5.8(d) illustrates that SVM performs the best on the class "Mention." However, it performs poorly on all other classes, because it classifies a majority of instances into "Mention" class.

5.5.3 Case Study

To investigate the advantages of multi-task classification in details, we look at a few sample tweets and their classification results with different methods. Tables 5.2 and 5.3 show 4 sample tweets with their sentiment and topic classification labels. Besides the ground truth label in the Truth column, we list classification results by NB, SVM, ME and our MTML model.

Case 1. *Tweet #1 has topic Complaint and sentiment Negative. NB, SVM and ME all classify it to the wrong topic and wrong sentiment classes. However, by us-*

ing the multi-task approach and incorporating the association between Complaint and Negative, our MTML model successfully classifies it to the right topic and sentiment. ■

Case 2. *Tweet #2 has topic Compliment and sentiment Positive. Keyword "love" is a strong indicating feature, but neither SVM nor ME classifies it right. MTML introduces multi-task and multi-label based on ME, therefore, MTML generates the correct classification results.* ■

Case 3. *Tweet #3 has topic Promotion and sentiment Positive. Both ME and SVM fail to classify on topic or sentiment. NB classifies with only right sentiment. As a comparison, MTML benefits from multi-task and makes right classifications on both tasks.* ■

Case 4. *Tweet #4 has topic Mention and sentiment Neutral. Among baselines, only SVM classifies its topic correctly. NB classifies with an incorrect association between topic and sentiment. ME does not classify correctly on either task. Only MTML utilizes multi-task labels to promote each other, and successfully classifies both topic and sentiment accurately.* ■

The above experiment shows that the MTML model performs better than baseline methods on both sentiment and topic classification. It produces classification accuracies of 0.744 on sentiment and 0.558 on topic. Compared with ME, MTML improves the accuracy by 5% on sentiment and 12% on topic classification, which indicates that using multi-label and multi-task is effective to improve both classifications. In particular, topic classification obtains a higher accuracy increase than sentiment classifications. It appears that incorporating sentiment labels seems to

be of more help to distinguish topics. Looking into accuracies per each class also reveals some insights. Among all classes, for instance, MTML has a higher accuracy on large-sized ones, such as “Complaint”, “Mention”, and “Promotion.” Since topic classes have unbalanced distributions and some of them have very few instances, increasing the dataset size may help increase the classification accuracy.

Chapter 6

Mining Impact of Events from Twitter Stream

Based on social activity temporal prediction and social media sentiment analysis, we furthermore explore how the evolution of social activity can help with predicting the sentiment change of Twitter stream. In this chapter, based on aggregate social activity, we utilize a continuous-time stochastic model to simulate and predict the sentiment change of Twitter stream. Therefore, analysis of tweet sentiment change can provide insights to the impact of events.

6.1 Literature Survey

6.1.1 Sentiment Analysis of Tweets

Tweet sentiment analysis focuses on identifying tweet sentiments from tweet contents, hashtags and emoticons [75]. Beside the ones mentioned in the previous section, there are more related work on tweet sentiment analysis. In [76], tf-idf

measure is used to detect the change of term frequency and emoticons are used to determine tweet sentiment. Another study proposed an algorithm by using recursive autoencoder to analyze sentiments of tweets [77]. To explore the tweet sentiment change on time-series, SVM was used to classify whether the collective tweet sentiment would increase or decrease [78]. After that, a study on sentiment of tweets by popular users found that these tweets had influence on the sentiment of their audience [79]. In [80], a method is proposed to train a sentiment analysis model with manually labeled data and emoticon labels are used to enhance the accuracy of classification.

Although these works conduct various analysis of tweet sentiment, none of them makes prediction of tweet sentiment change.

6.1.2 Electoral Prediction with Twitter

In the realm of electoral prediction, a lot of studies analyze tweet sentiment and seek to reveal hidden information from the results [81, 82]. In [83], positive and negative scores are integrated to compute sentiment scores of tweets, which are found to be relevant to the presidential approval polls. After that, researchers found that mere count of tweets mentioning a party or candidate can reflect the election results [84]. In another study, demographic information of twitter users are analyzed and some conclusions are drawn, such as users are predominantly male [85]. Besides that, the influence of vocal minority is also examined. Compared to silent majority, vocal minority is found to play a major role in spreading information aligned with their own opinions [86]. By analyzing online popularity of Italian political leaders in 2011 and online voting intension of French 2012 election, researchers confirmed a remarkable ability for social media to forecast

electoral results [87].

On the other hand, some studies raised doubts and pointed out insufficiency of predicting elections with twitter [88]. In [89], 2011 Singapore General Election is analyzed, but results show that the correlation between Twitter chatter and votes is not strong enough to make accurate predictions. By examining tweets about US republican politicians in 2011 US presidential nomination, another study also shows that twitter political chatter is not indicative of national political polls [90].

Upon the above conclusions of electoral prediction with twitter, most of them are not replicable. The prediction is mostly based on qualitative analysis or counting the number of tweets, instead of quantification of impact of events. Therefore, in this thesis, we present a new method that measures the impact of events, which furthermore can predict the change of tweet sentiments.

6.2 Dataset

By using Twitter Search API, we collected tweets mentioning presidential candidates of 2012 US presidential campaign. Each tweet contains either "obama" or "romney". The main Twitter stream is called Firehose and one subsample of this stream is named Gardenhose. With Gardenhose access, the program is allowed to access at most 10% of the main Twitter stream. Filtering the stream with special keywords will furthermore reduce the percentage of tweets that we can collect.

The collection of tweets spans from March 23, 2012 to November 10, 2012, with a few gaps of uncollected time intervals. As the stream connection dropped from time to time, this dataset is lack of some days in the period. The tweets are saved in JSON format. The total size of the dataset is around 140GB.

6.2.1 Tweet Sentiment Analysis

The classification scheme of tweet sentiment are: *positive*, *negative* and *neutral*. To identify tweet sentiment in the huge dataset collection, we use Maximum Entropy as a supervised classification model. A subset of 10,000 tweets are used as training instances. Then we make use of Amazon Mechanical Turk(AMT) to obtain sentiment labels. As introduced in the previous chapter, AMT is a crowdsourcing marketplace which allows collaboration of people to complete intelligent tasks. By using AMT we collect sentiment label for every training tweet.

After that, a sentiment classification model is trained with Maximum Entropy. Considering that sentiment analysis is very subjective, the short length of a tweet may make it even harder to correctly identify tweet sentiment. Shown by experiment, the model produces 77% classification accuracy when 200 instances are used for testing and the rest used for training. We consider it a reasonable classification accuracy, therefore we apply this model to identify the sentiment of all other tweets in the collection.

6.2.2 Social Activity Feature Extraction

With tweet sentiment labels obtained, we can make use of them and extract social activity features for prediction. To measure user activity and their interactions, we set the time interval to be 12 hours and partition the dataset accordingly. On every 12-hour interval, tweets about each politician are separated. Given tweet sentiment labels, 30 activity features are generate for each politician on every interval. Table 6.1 shows the feature sources and their explanations.

Since the value of most features evolve significantly over time, all features except

Table 6.1. Social Activity Features from Twitter User Network

Sources	Example Feature and Explanation
Users	Number of followers Number of friends Number of posted tweets Number of lists the user is in
Sentiment	Number of positive tweets Number of negative tweets Number of neutral tweets
Tweets	Number of retweets Number of users who have positive tweets Number of users who have negative tweets Number of users who have neutral tweets
Historical change	First order derivative of sentiment features and tweet features

for the first order derivative are rescaled as follows to handle the very large values:

$$f^* = log(f + 1) \tag{6.1}$$

where f is the original value of the feature and f^* is the new value after the rescaling.

These features together will be used as model input to predict the impact of events on the change of tweet sentiments towards each politician.

6.3 Temporal Sentiment Analysis

To analyze the temporal pattern of the tweets, we sort the collection with temporal order, and then separate them according to the political leader mentioned. Figure 6.1 shows the weekly sum of tweet numbers about Obama and Romney, respectively. Overall, the tweet number has a tendency of increasing over time,

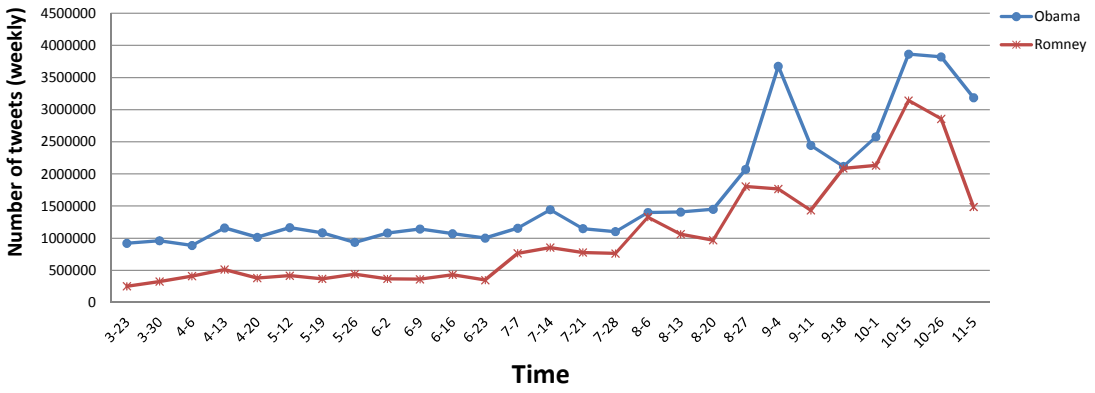


Figure 6.1. Nationwide weekly tweet numbers of Obama and Romney

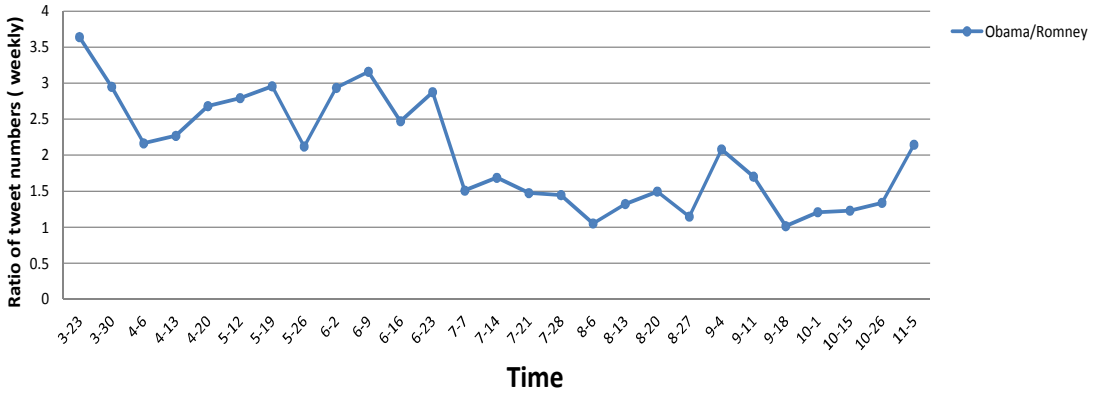


Figure 6.2. Ratio of nationwide weekly tweet numbers of Obama and Romney

from July 2012 to October 2012. There are more tweets talking about Obama than Romney all the time. Figure 6.2 shows the ratio of weekly tweet numbers of Obama and Romney. From March 2012 to November 2012, the ratio keeps decreasing over time generally.

An interesting observation is that during the two weeks starting with August 27th, the number of tweet about Romney has a big increase, followed by another big increase of tweets about Obama in the next week. The happening at these two weeks are the Republican national convention from August 27th to August 30th, and Democratic national convention from September 3rd to September 6th. After

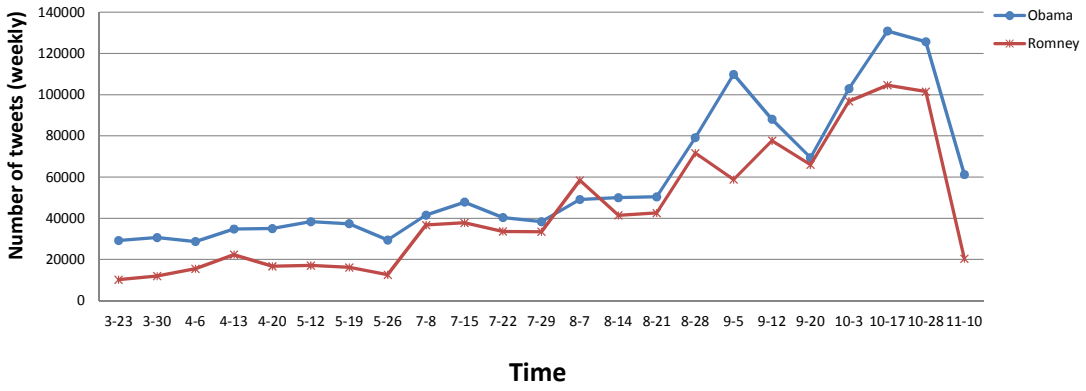


Figure 6.3. Weekly tweet numbers of Obama and Romney in CA

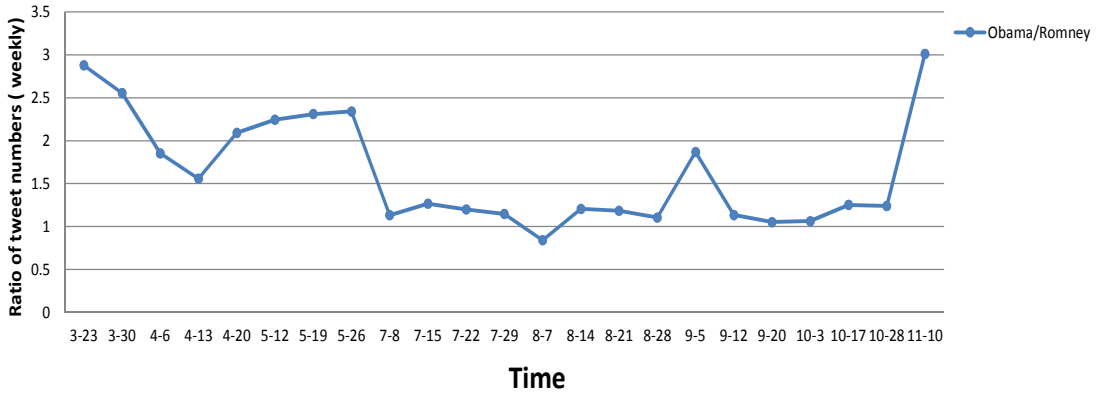


Figure 6.4. Ratio of weekly tweet numbers of Obama and Romney in CA

that, tweets of both politicians start to increase dramatically. It implies a strong correlation between the political event and the increasing tweets of the politicians. After the final election on November 6th and the win of Obama, the tweet number of two politicians both decrease, but the ratio shows a big increase. It can be explained as that the winner of election draws much more attention again.

Figure 6.3 shows the weekly sum of tweets in California about the two politicians, while figure 6.4 illustrates the ratio between them. The plots depict a similar temporal pattern as that of nationwide. However, the number of tweets about Romney is closer to that of Obama than national data since July. It indicates that

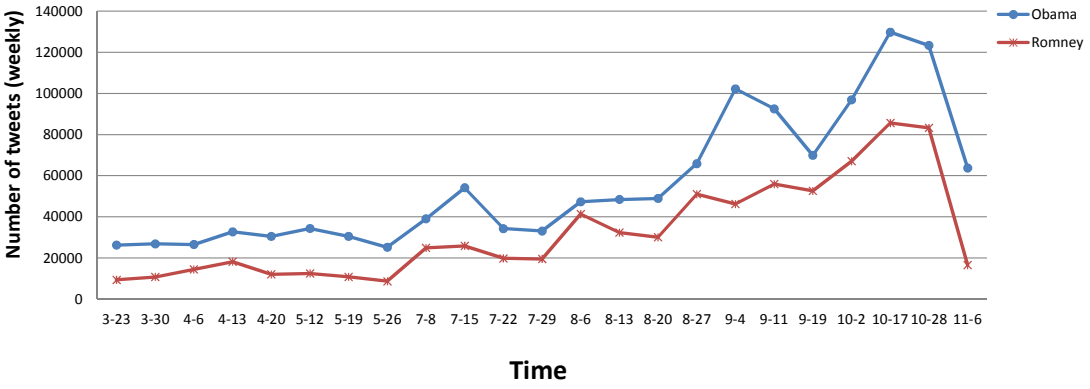


Figure 6.5. Weekly tweet numbers of Obama and Romney in TX

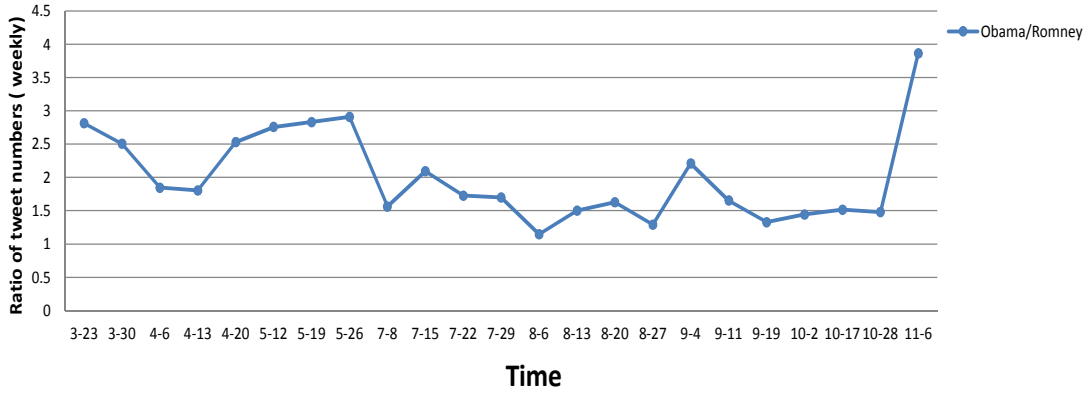


Figure 6.6. Ratio of weekly tweet numbers of Obama and Romney in TX

Romney is mentioned more frequently by California Twitter users than national average.

Figure 6.5 and figure 6.6 show the weekly sum of tweets and the ratio of Obama and Romney in Texas. Comparing figure 6.4 and figure 6.6, we observe that the ratio of tweets about Obama and Romney in Texas is higher than that in California. It indicates that Twitter users from a strong republican state, such as Texas, comment more about Obama than users from California and nationwide. Given this observation, we are curious what these Twitter users are talking about Obama and how their sentiment is. In the following analysis, we will look into the

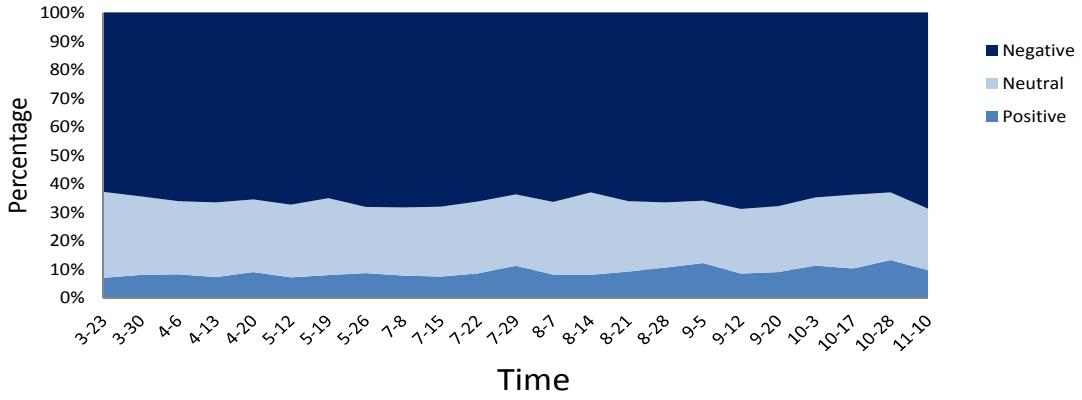


Figure 6.7. Distribution of tweets with different sentiments towards Obama in CA

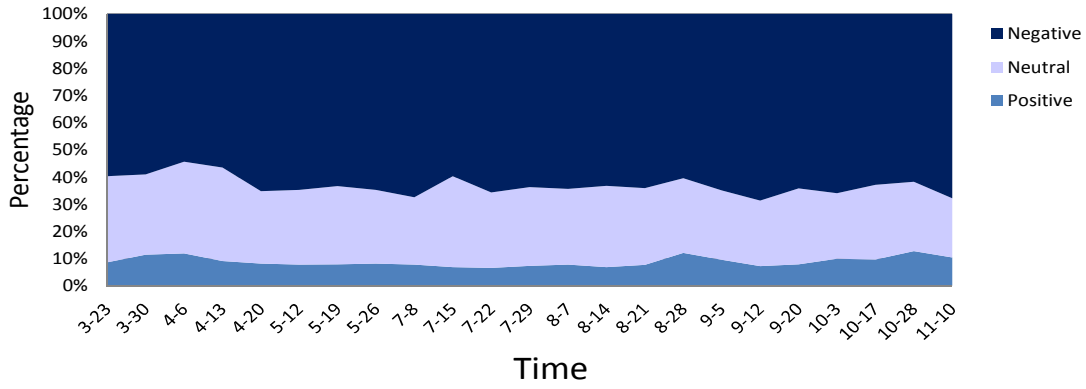


Figure 6.8. Distribution of tweets with different sentiments towards Romney in CA

tweet sentiment distribution and find out answer to this question.

By using the sentiment classification model, we classify sentiments of all tweets in the collection. Then we calculate the number of positive, negative, and neutral tweets about Obama and Romney in both California and Texas.

Figure 6.7 shows the distribution of tweets with different sentiments about Obama in California. Negative tweets take a proportion of more than half, and the proportion stays mostly at the same time from March to November. Positive tweets has a proportion of around 10%, but the proportion shows a increasing trend all over the time. It indicates that there are more positive tweets about

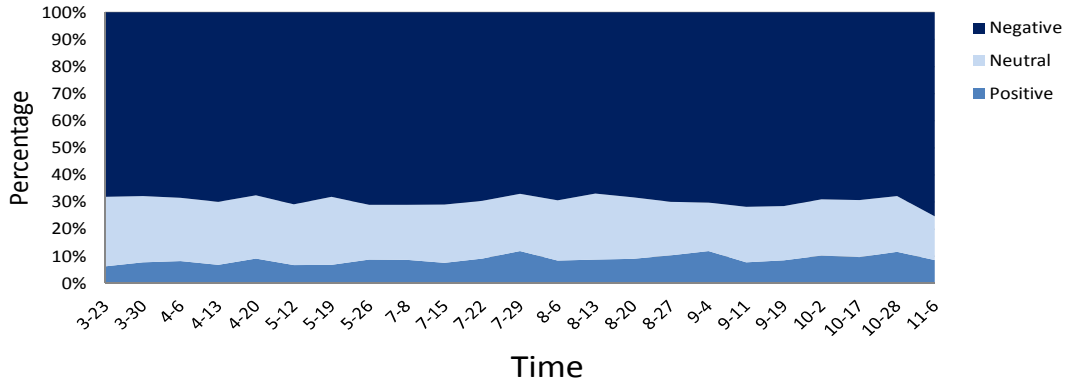


Figure 6.9. Distribution of tweets with different sentiments towards Obama in TX

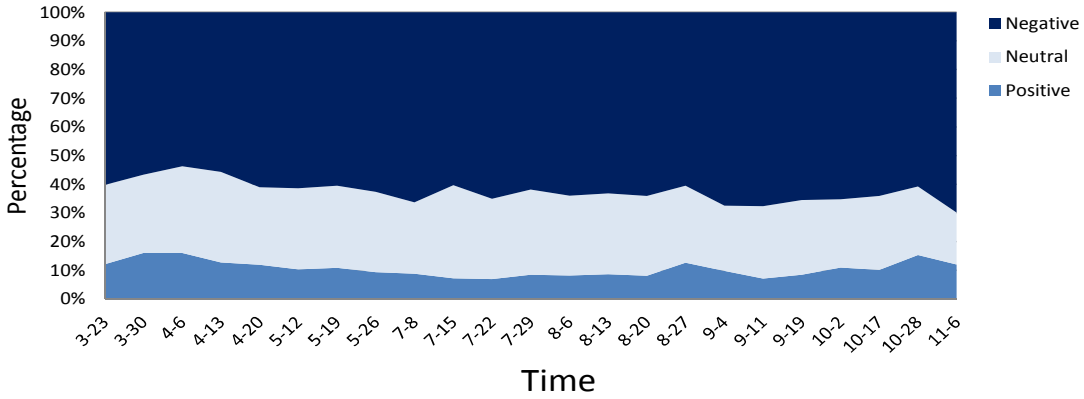


Figure 6.10. Distribution of tweets with different sentiments towards Romney in TX

Obama in California as the election is approaching.

Figure 6.8 shows the distribution about tweets toward Romney in California. In the figure, the percentage of positive and negative tweets are mostly stable, with a slightly increase of positive tweets after September. Comparing figure 6.7 and figure 6.8, we can observe that the sentiment distributions of the two politicians in California are almost even.

Figure 6.9and figure 6.10 shows the distribution about tweets of Obama and Romney in Texas. Comparing the two figures, on one hand, we find that the percentage of negative tweets about Obama is obviously higher than that of Rom-

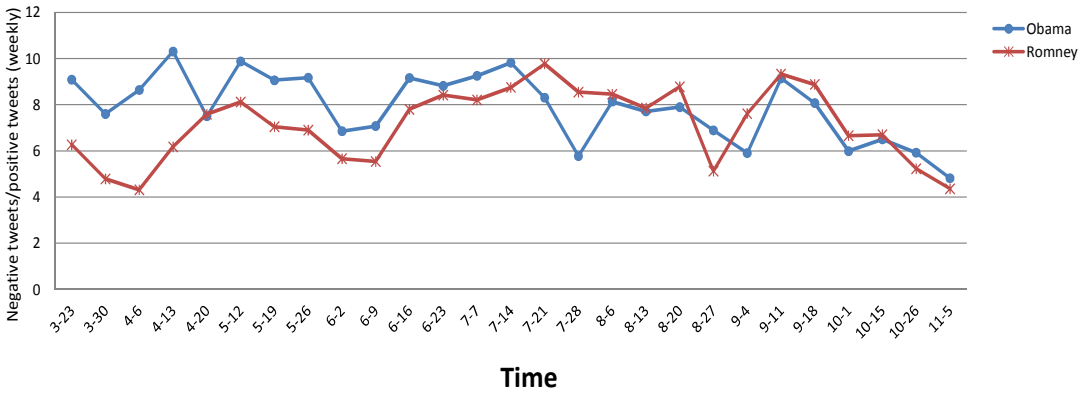


Figure 6.11. Nationwide ratio of negative and positive tweets for Obama and Romney

ney. On the other, the percentages of positive tweets about them are similar. It is an interesting observation that although Texas Twitter users twit more about Obama, he actually receives a higher percentage of negative comments than Romney in Texas. On the contrary, California Twitter users show a more even attitude towards both politicians.

Furthermore, we measure the ratio of negative and positive tweets about politicians. Figure 6.11 shows the nationwide ratios of Obama and Romney. From March to June, Obama has a higher ratio of negative tweets than Romney. Since July, ratios of the two become similar and the average sentiment towards Obama is improved.

Figure 6.12 and figure 6.13 show the ratio of negative and positive tweets in California and Texas, respectively. Since July, the plot of Romney starts to show a higher negative ratio than Obama in CA, and a slightly lower negative ratio in TX, especially after September. The final election result is that Obama wins in CA and Romney wins in TX, which agrees with our observations.

From all above observations and analysis, we consider that tweet sentiment classification can reflect the political preference of both national average and states.

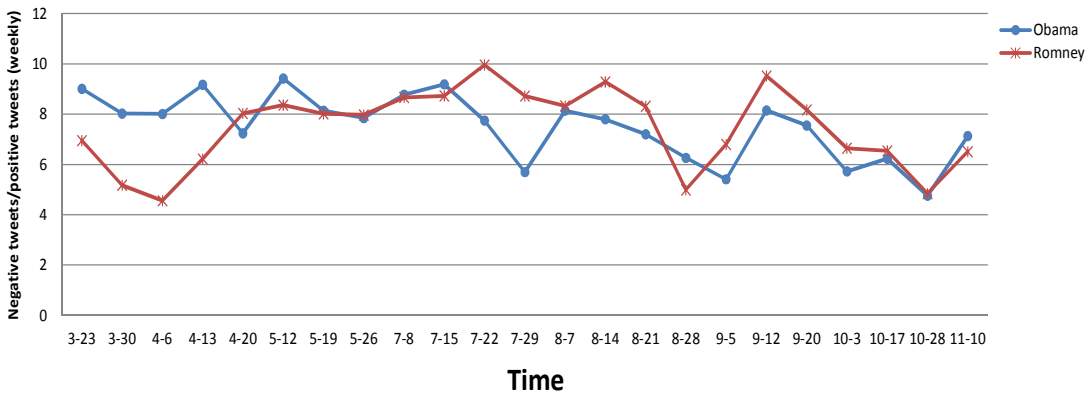


Figure 6.12. Ratio of negative and positive tweets for Obama and Romney in CA

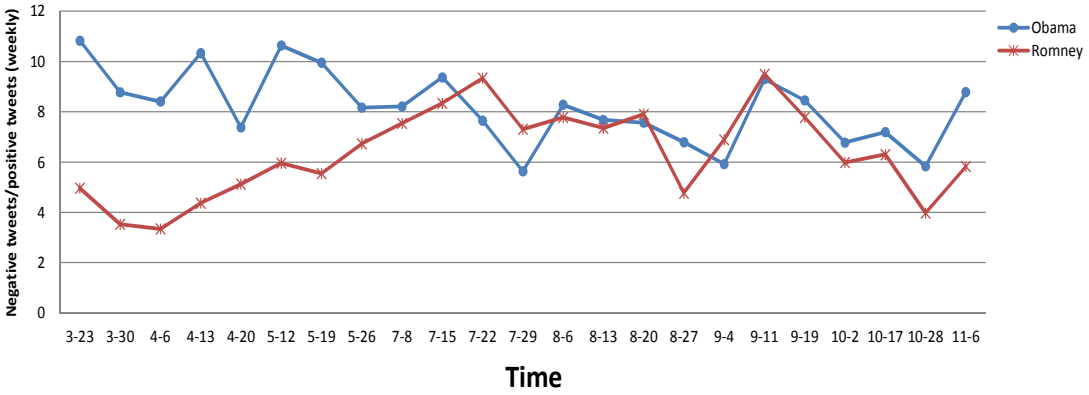


Figure 6.13. Ratio of negative and positive tweets for Obama and Romney in TX

Comparing the tweet statistics and the facts, we find that they mostly agree with each other. Therefore, we can conclude that tweet sentiment analysis is a credible approach to reveal some hints of the facts.

6.4 Sentiment Change Prediction

6.4.1 Methodology

Sentiment change of tweet can be measured by the number of tweets that express positive or negative sentiment. Therefore, the question of predicting such sentiment

change can be converted to predicting the future number of positive or negative tweets. Due the randomness in social dynamics, the temporal change of number of tweets is essentially a continuous-time random variable. Thus it can be properly simulated by a continuous-time stochastic process. In fact, the number of tweet is a reflect of social activity in Twitter user network.

Based on this observation, we adopt a Parameterized Social Activity Model (PSAM) [91] which can simulate the social activity evolution over continuous-time. As a simulation model, PSAM has two components: a drift term and a diffusion term. The drift term indicates the growth or shrinkage of the social activity. The diffusion term describes the uncertainty, e.g. the impact from the environment. Having these two components integrated into it, PSAM can simulate and predict the evolution of social activity accurately. Therefore, we utilize it to predict the change of number of tweets.

6.4.2 Experiments and Discussion

The dataset is partitioned to two parts for experimental validation. Tweets from March 23, 2012 to September 4, 2012 are used for training, and the part from September 5, 2012 to November 10, 2012 is used for testing. We measure the change of both positive and negative tweets.

First, by using PSAM, we predict the number of positive/negative tweets towards each politician. Then we measure the accuracy of 90% prediction interval(PI) and correlation coefficient between the predicted value and the ground truth.

Ground truth: With every tweet labeled as positive, negative, or neutral, the change rate of positive tweets is calculated with number of positive tweets on

Table 6.2. Accuracy of 90% PI of prediction on positive and negative tweets

90% PI	Positive	Negative
Obama	0.63	0.57
Romney	0.88	0.87

Table 6.3. Correlation Coefficient of predictions and ground truth on positive and negative tweets

Correlation	Positive	Negative
Obama	0.522	0.609
Romney	0.7	0.7

every pairs of two successive 12-hour intervals. The change rate of negative tweets is calculated in the same way.

Confidence interval: We calculate the prediction interval (PI) from the predicted distribution and compare the ground truth with it. To ensure a tight interval and enough confidence, we adopt the 90% PI. Suppose $\Re$ represents the set of testing data and $\mathfrak{T}$ is the subset that falls within 90% PI, then the 90% PI accuracy is:

$$Accuracy_{90\%PI} = \frac{|\mathfrak{T}|}{|\Re|}$$

Correlation coefficient: Correlation coefficient is a widely accepted approach to measure the relevance between estimated values and ground truth. In particular, assume that the ground truth dataset is X and corresponding predicted dataset is Y, $\bar{x}$ and $\bar{y}$ denotes the mean values of X and Y respectively, then the correlation between X and Y is calculated as follows:

$$Corr(X,Y) = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2 \sum(y-\bar{y})^2}}$$

Table 6.2 shows accuracy of 90% PI of predictions for Obama and Romney. For each politician, the prediction accuracies of positive tweets and negative tweets

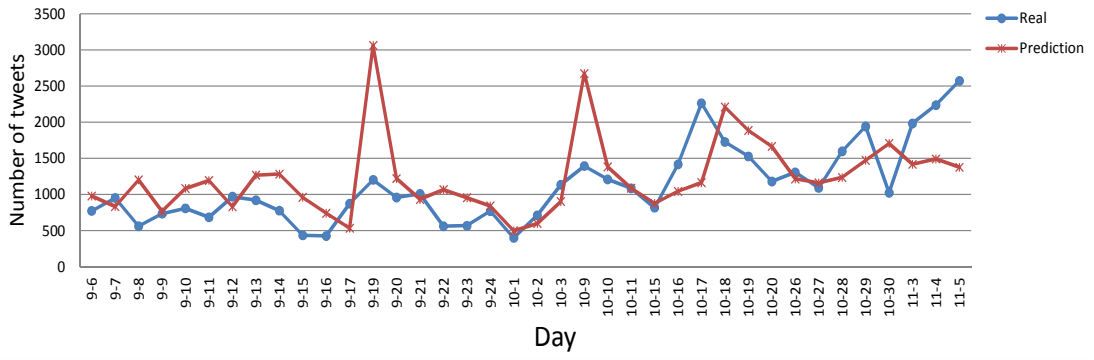


Figure 6.14. Daily positive tweet numbers of Romney, prediction VS ground truth

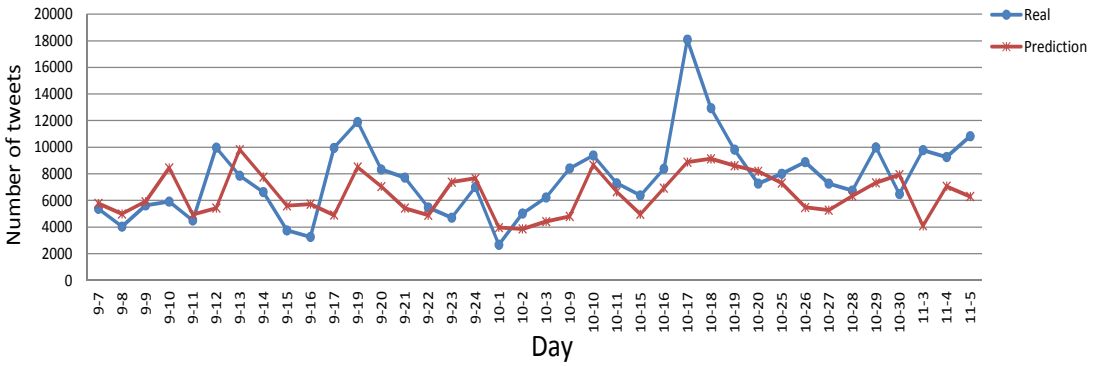


Figure 6.15. Daily negative tweet numbers of Romney, prediction VS ground truth

are close. However, prediction accuracies of Romney is much higher than those of Obama. Table 6.3 illustrates correlation coefficients of predictions and the ground truth. Predictions about both candidates all have a high correlation, which implies that sentiment change is predicted accurately. Meanwhile, we also notice that predictions about Romney have higher correlations than those about Obama, which is according with the results in Table 6.2.

One possible reason of the less accurate predictions about Obama could be that some tweets contains keyword "Obama" may actually talk about Michelle Obama, which lead to more noise in the dataset. Meanwhile, since Obama is the current president, his name could be mentioned upon many issues. Tweets containing "Obama" may not be relevant the presidential campaign.

During the presidential campaign, each candidate has a lot of public activities to call for more votes. Presidential debate is one of them. As presidential debates are broadcasted on TV channels, they make wide national influence. Therefore, to validate our methodology, we look into the daily tweet sentiment change about Romney in California, especially after the presidential debate on October 16th.

Figure 6.14 shows the real and predicted number of positive tweets about Romney in CA. Overall, the prediction catches the up and down oscillation of ground truth, with a slight delay on a few days. On days after October 16th, the model successfully predicts the increase of positive tweet number, though the peak is predicted with one day shift. Also the prediction is not far from the ground truth.

Figure 6.15 shows the real and predicted number of negative tweets of Romney on the same days. The prediction has a good match to the ground truth. Focusing on days after October 16th, PSAM makes an accurate prediction of negative tweet increase. In fact, in the debate of October 16th, Romney received a very negative feedback. Considering figure 6.14 and figure 6.15 together, according to the model predictions, negative tweet number has a much more increase than positive tweet. This result accurately reflects the fact and properly predicts the impact of the presidential debate.

Overall, we can conclude that predicting the impact of events on social media with PSAM is accurate and efficient. Analyzing UGC provides some insights into the current status of social communities. By looking into the temporal social interactions, our method is able to catch the influence of events on social activities and therefore reveal the future impact in the community, in particular, fluctuation of sentiments.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this dissertation, we study UGC on the web and social networks and utilize the results to improve the solutions to several challenging data management problems. First, a statistic model incorporating document topics and user locations is proposed to improve query expansion. Second, we explore various social activity features and evaluate which structural features are important in determining community evolution. Third, we analyze sentiments and topics of UGC in social media and propose a multi-task multi-label classification model. Furthermore, based on UGC sentiment analysis, we present a method that can predict sentiment change on social media and forecast impact of events.

In the problem of query expansion, search engine user log is utilized to help explore the semantic correlation of searching documents. By clustering documents into different topics, we scale down the document relevance to the topic relevance with LDA, and then use the topic relevance to identify the similarity between queries. In addition we make use of the location information to determine whether

the query is location-sensitive and which type of query expansion should be applied. Our experiments on the CiteSeer and Excite datasets show that on one hand, our model can effectively select the location-sensitive queries; on the other hand, for location-sensitive queries, our query expansion methods significantly improve the search results.

To address the second problem, we focus on investigating the impact of member interaction over the active community evolution from a macro scope. Observing the temporal network infrastructure, we find that network growth and shrinkage tend to be consistent with the number of active members and the interaction between them. Therefore, we formalize the concept of active social network and make use of evolving patterns to measure community evolution. Several structural features are incorporated to represent member activities, and then they are applied with the logistic regression to predict the evolving pattern. At the same time, the Lasso method is adopted to select the most significant structural features. The experiment on both CiteSeer co-authorship network and Facebook online network shows that our methods are effective in predicting the evolving pattern accurately and that the feature selection is valid. The most significant structural features selected are different on the two datasets. On the Facebook online social network, the numbers of current members and cumulative edges are more important than other factors. On the CiteSeer co-authorship network, the collaboration between members plays the most important role.

In the third application, we study the sentiment and topic classification of online posts. By exploring the latent association between tweet sentiments and topics, we propose a multi-task multi-label (MTML) classification model. The model utilizes the correlation between related classes across two tasks, and incor-

porates the result of each classification task to promote the other. In addition, the MTML model integrates multi-label in training to learn from ambiguous expressions and to classify such accordingly. Experiments on a collection of real tweets using crowdsourced ground truth reveal that our proposed model can classify both sentiments and topics of tweets accurately and outperforms other four competing methods.

Furthermore, based on sentiment analysis of online posts, the dynamic correlation between social activity and UGC is investigated. By using a parameterized social activity model, we explore the change of sentiment expressed in online posts and utilize it to predict the impact of events. Experiments show that our method can predict the sentiment change accurately. From analysis and case study, we find that mining temporal social interaction can reveal the change of sentiment in UGC, and therefore, help with predicting the influence of events.

7.2 Future Research

Many research questions remain open for future work. Social activity and UGC are dynamically involving and have influence on each other. Social interactions promote the generation of UGC. Therefore, social activity can help predict UGC, while UGC can be used to understand the social community status. This interactive relationship can be investigated and studied furthermore in multiple perspectives.

As we have been studying social activity at the macro level, the individual level analysis would be another direction of extension. Connections and influential individuals play an important role in determining activity of the entire social network. Therefore, mining these features may provide different quantification of

social network evolution.

Different formats of UGC can be explored and exploited. In this thesis, we only look into text and document. The scope of our study may extend to other formats, including image, audio, and video. By analyzing new formats of UGC and integrating the techniques of text mining, we can address many other data management problems and improve solutions.

Bibliography

[1] Bollegala, D., Y. Matsuo, and M. Ishizuka (2009) "Measuring the similarity between implicit semantic relations using web search engines," in *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, ACM, New York, NY, USA, pp. 104–113.

[2] Bollegala, D. T., Y. Matsuo, and M. Ishizuka (2010) "Relational duality: unsupervised extraction of semantic relations between entities on the web," in *Proceedings of the 19th international conference on World wide web*, WWW '10, ACM, New York, NY, USA, pp. 151–160.

[3] Harrington, B. (2007) "ASKNet: automated semantic knowledge network," in *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*, AAAI Press, pp. 1862–1863.

[4] Dolby, J., A. Fokoue, A. Kalyanpur, A. Kershenbaum, E. Schonberg, K. Srinivas, and L. Ma (2007) "Scalable semantic retrieval through summarization and refinement," in *Proceedings of the 22nd national conference on Artificial intelligence - Volume 1*, AAAI Press, pp. 299–304.

[5] Liu, D., X.-S. Hua, M. Wang, and H.-J. Zhang (2010) "Retagging social images based on visual and semantic consistency," in *Proceedings of the 19th international conference on World wide web*, WWW '10, ACM, New York, NY, USA, pp. 1149–1150.

[6] Gracia, J., M. d'Aquin, and E. Mena (2009) "Large scale integration of senses for the semantic web," in *Proceedings of the 18th international conference on World wide web*, WWW '09, ACM, New York, NY, USA, pp. 611–620.

[7] Cudré-Mauroux, P., P. Haghani, M. Jost, K. Aberer, and H. De Meer (2009) "idMesh: graph-based disambiguation of linked data," in *Proceedings of the 18th international conference on World wide web*, WWW '09, ACM, New York, NY, USA, pp. 591–600.

[8] Wu, G., M. Yang, K. Wu, G. Qi, and Y. Qu (2010) "Falconer: once SIOC meets semantic search engine," in *Proceedings of the 19th international conference on World wide web*, WWW '10, ACM, New York, NY, USA, pp. 1317–1320.

[9] Lin, C., J.-M. Yang, R. Cai, X.-J. Wang, and W. Wang (2009) "Simultaneously modeling semantics and structure of threaded discussions: a sparse coding approach and its applications," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, ACM, New York, NY, USA, pp. 131–138.

[10] Baeza-Yates, R. and B. Ribeiro-Neto (1999) *Modern Information Retrieval*, Addison-Wesley Longman.

[11] Billerbeck, B., F. Scholer, H. E. Williams, and J. Zobel (2003) "Query Expansion using Associated Queries," in *Proc. of 12th International Conference on Information and Knowledge Management (CIKM'03)*, New Orleans, Louisiana, USA.

[12] Cai, D., C. J. van Rijsbergen, and J. M. Jose (2001) "Automatic Query Expansion Based on Divergence," in *Proc. of 10th International Conference on Information and Knowledge Management (CIKM'01)*, Atlanta, Georgia, USA.

[13] Park, L. a. F. and K. Ramamohanarao (2004) "Hybrid Prequery Term Expansion Using Latent Semantic Analysis," in *Proc. of the 4th International Conference on Data Mining (ICDM'04)*.

[14] Blei, D. M., A. Y. Ng, and M. I. Jordan (2003) "Latent Dirichlet Allocation," *Machine Learning Research*, **3**, pp. 993–1022.

[15] Sun, J., C. Faloutsos, S. Papadimitriou, and P. S. Yu (2007) "GraphScope: parameter-free mining of large time-evolving graphs," in *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, New York, NY, USA, pp. 687–696.

[16] Faloutsos, M., P. Faloutsos, and C. Faloutsos (1999) "On power-law relationships of the Internet topology," in *SIGCOMM '99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pp. 251–262.

[17] Barabasi, A.-L. and R. Albert (1999) "Emergence of Scaling in Random Networks," *Science*, **286**, pp. 509–512.

[18] Asur, S., S. Parthasarathy, and D. Ucar (2007) "An event-based framework for characterizing the evolutionary behavior of interaction graphs," in *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, New York, NY, USA, pp. 913–921.

[19] Wang, X., F. Wei, X. Liu, M. Zhou, and M. Zhang (2011) "Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach," in *ACM CIKM*, pp. 1031–1040.

[20] Jiang, L., M. Yu, M. Zhou, X. Liu, and T. Zhao (2011) "Target-dependent Twitter sentiment classification," in *ACL*, pp. 151–160.

[21] Tan, C., L. Lee, J. Tang, L. Jiang, M. Zhou, and P. Li (2011) "User-level sentiment analysis incorporating social networks," in *ACM KDD*, pp. 1397–1405.

[22] Agarwal, A., B. Xie, I. Vovsha, O. Rambow, and R. Passonneau (2011) "Sentiment analysis Twitter data," in *Workshop on Languages in Social Media*, pp. 30–38.

[23] Nishida, K., T. Hoshide, and K. Fujimura (2012) "Improving tweet stream classification by detecting changes in word probability," in *ACM SIGIR*.

[24] Yu, S. and S. Kak (2012) "A Survey of Prediction Using Social Media," *CoRR*, **abs/1203.1647**.

[25] Arasu, A., J. Cho, H. Garcia-Molina, A. Paepcke, and S. Raghavan (2001) "Searching the web," *ACM Transactions on Internet Technology (TOIT)*, **1**(1), p. 43.

[26] Collins-Thompson, K. and J. Callan (2005) "Query Expansion Using Random Walk Models," in *Proc. of 14th International Conference on Information and Knowledge Management (CIKM'05)*, Bremen, Germany.

[27] Cao, G., J.-Y. Nie, and J. Bai (2005) "Integrating Word Relationships into Language Models," in *Proc. of the ACM Conference on Research and Development in Information Retrieval (SIGIR'05)*, Salvador, Brazil.

[28] Bai, J., D. Song, P. Bruza, J.-Y. Nie, and G. Cao (2005) "Query Expansion Using Term Relationships in Language Models for Information Retrieval," in *Proc. of 14th International Conference on Information and Knowledge Management (CIKM'05)*, Bremen, Germany.

[29] Cui, H., J.-R. Wen, J.-Y. Nie, and W.-Y. Ma (2002) "Probabilistic Query Expansion Using Query Logs," in *Proc. of 11th international Conference on World Wide Web (WWW'02)*, Honolulu, Hawaii, USA.

[30] Xue, G.-R., H.-J. Zeng, Z. Chen, Y. Yu, W.-Y. Ma, W. Xi, and W. Fan (2004) "Optimizing Web Search Using Web Click-through Data," in *Proc. of 13th International Conference on Information and Knowledge Management (CIKM'04)*, Washington, DC, USA.

[31] Fonseca, B. M., P. Golgher, and B. Pôssas (2005) "Concept-Based Interactive Query Expansion," in *Proc. of 14th International Conference on Information and Knowledge Management (CIKM'05)*, Bremen, Germany.

[32] Kendall, M. G. (1995) *Rank Correlation Methods, Second Edition*, New York: Hafner Publishing Co.

[33] Agichtein, E., E. Brill, and S. Dumais (2006) "Improving web search ranking by incorporating user behavior information," in *Proc. of the ACM Conference on Research and Development in Information Retrieval (SIGIR'06)*, Seattle, USA.

[34] Chang, C.-C. and C.-J. Lin (2007) *LIBSVM: a Library for Support Vector Machines*.

[35] Mitra, P. and G. Wiederhold (2002) "Resolving Terminological Heterogeneity in Ontologies," in *Proc. of Workshop on Ontologies and Semantic Interoperability at the 15th European Conference on Artificial Intelligence (ECAI'02)*.

[36] Leskovec, J., K. J. Lang, A. Dasgupta, and M. W. Mahoney (2008) "Statistical properties of community structure in large social and information networks," in *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pp. 695–704.

[37] Kumar, R., J. Novak, and A. Tomkins (2006) "Structure and evolution of online social networks," in *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 611–617.

[38] Watts, D. J. and S. H. Strogatz (1998) "Collective dynamics of 'small-world' networks," *Nature*, **393**(6684), pp. 440–442.

[39] Albert, R., H. Jeong, and A.-L. Barabasi (1999) "Diameter of the World-Wide Web," *Nature*, **401**, pp. 130–131.

[40] Newman, M. E. J. and J. Park (2003) "Why social networks are different from other types of networks," *Physical Review E*, **68**, p. 36122.

[41] Erdös, P. and A. Rényi (1959) "On Random Graphs," *Publicationes Mathematicae Debrecen*, **6**, p. 290.

[42] Elmacioglu, E. and D. Lee (2009) "Modeling idiosyncratic properties of collaboration networks revisited," *Scientometrics*, **80**(1), pp. 195–216.

[43] Leskovec, J., L. Backstrom, R. Kumar, and A. Tomkins (2008) "Microscopic evolution of social networks," in *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 462–470.

[44] Mislove, A., M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee (2007) "Measurement and analysis of online social networks," in *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pp. 29–42.

[45] Leskovec, J., J. Kleinberg, and C. Faloutsos (2005) "Graphs over time: densification laws, shrinking diameters and possible explanations," in *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 177–187.

[46] Ahn, Y.-Y., S. Han, H. Kwak, S. Moon, and H. Jeong (2007) "Analysis of topological characteristics of huge online social networking services," in *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pp. 835–844.

[47] Zheleva, E., H. Sharara, and L. Getoor (2009) "Co-evolution of social and affiliation networks," in *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1007–1016.

[48] Lin, Y.-R., Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng (2008) "Facetnet: a framework for analyzing communities and their evolutions in dynamic networks," in *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pp. 685–694.

[49] Palla, G., A.-L. Barabasi, and T. vicsek (2007) "Quantitative social group dynamics on a large scale," *Nature*.

[50] Backstrom, L., D. Huttenlocher, J. Kleinberg, and X. Lan (2006) "Group formation in large social networks: membership, growth, and evolution," in *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 44–54.

[51] Tibshirani, R. (1996) "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society (Series B)*, **58**, pp. 267–288.

[52] Viswanath, B., A. Mislove, M. Cha, and K. P. Gummadi (2009) "On the evolution of user interaction in Facebook," in *WOSN '09: Proceedings of the 2nd ACM workshop on Online social networks*, pp. 37–42.

[53] Giles, C. L., K. D. Bollacker, and S. Lawrence (1998) "Citeseer: an automatic citation indexing system," in *Prodeedings of the third ACM conference on Digital libraries*.

[54] J.Han and M.Kanber (2006) in *Data Mining: Concepts and Techniques, Second Edition*, pp. 302–304.

[55] Boutell, M. R., J. Luo, X. Shen, and C. M. Brown (2004) "Learning multi-label scene classification," *Pattern Recognition*, pp. 1757–1771.

[56] Tai, F. and H.-T. Lin (2012) "Multilabel classification with principal label space transformation," *Neural Comput.*, **24**, pp. 2508–2542.

[57] Lauser, B. and A. Hotho (2003) "Automatic multi-label subject indexing in a multilingual environment," in *7th European Conf. in Research and Advanced Technology for Digital Libraries*, pp. 140–151.

[58] Tsoumakas, G. and I. Katakis (2007) "Multi-label Classification: An Overview," *Int J. Data Warehousing and Mining*, **3**, pp. 1–13.

[59] McCallum, A. K. (1999) "Multi-label text classification with a mixture model trained by EM," in *AAAI Workshop on Text Learning*.

[60] Clare, A. and R. D. King (2001) "Knowledge Discovery in Multi-label Phenotype Data," in *5th European Conf. on Principles Data Mining and Knowledge Discovery*, pp. 42–53.

[61] Thabtah, F. A., P. Cowling, and Y. Peng (2004) "MMAC: A New Multi-class, Multi-label Associative Classification Approach," in *IEEE ICDM*, pp. 217–224.

[62] Jin, R. and Z. Ghahramani (2003) "Learning with Multiple Labels," in *Conf. on Neural Information Processing Systems (NIPS)*.

[63] Caruana, R. (1997) "Multitask Learning," *Mach. Learn.*, **28**, pp. 41–75.

[64] Evgeniou, O. and M. Pontil (2004) "Regularized multi–task learning," in *ACM KDD*, pp. 109–117.

[65] Evgeniou, O., C. A. Micchelli, and M. Pontil (2005) "Learning Multiple Tasks with Kernel Methods," *J. Machine Learning Research*, **6**, pp. 615–637.

[66] Chapelle, O., P. Shivaswamy, S. Vadrevu, K. Weinberger, Y. Zhang, and B. Tseng (2010) "Multi-task learning for boosting with application to web search ranking," in *ACM KDD*, pp. 1189–1198.

[67] Ben-David, S. and R. Schuller (2003) "Exploiting Task Relatedness for Mulitple Task Learning," in *Annual Conf. on Computational Learning Theory*, pp. 567–580.

[68] Jebara, T. (2004) "Multi-task feature and kernel selection for SVMs," in *Int'l Conf. on Machine learning (ICML)*.

[69] Argyriou, A., odoros Evgeniou, and M. Pontil (2006) "Multi-Task Feature Learning," in *Conf. on Neural Information Processing Systems (NIPS)*, pp. 41–48.

[70] Xue, Y., X. Liao, L. Carin, and B. Krishnapuram (2007) "Multi-Task Learning for Classification with Dirichlet Process Priors," *J. Machine Learning Research*, **8**, pp. 35–63.

[71] Chua, F. C. T., W. W. Cohen, J. Betteridge, and E.-P. Lim (2012) "Community-Based Classification Noun Phrases in Twitter," in *ACM CIKM*.

[72] Nishida, K., R. Banno, K. Fujimura, and T. Hoshide (2011) "Tweet classification by data compression," in *Int'l Workshop on DETecting and Exploiting Cultural diversiTy on social web*, pp. 29–34.

[73] Lee, K., D. Palsetia, R. Narayanan, M. M. A. Patwary, A. Agrawal, and A. Choudhary (2011) "Twitter Trending Topic Classification," in *IEEE IDCM Workshops*, pp. 251–258.

[74] McCallum, A. K. (2002) "MALLET: A Machine Learning for Language Toolkit," Http://mallet.cs.umass.edu.

[75] Kouloumpis, E., T. Wilson, and J. Moore (2011) "Twitter Sentiment Analysis: The Good the Bad and the OMG!" *Proc. of the Fifth International AAAI Conference on Weblogs and Social Media*.

[76] Bifet, A., G. Holmes, B. Pfahringer, and R. Gavalda (2011) "Detecting Sentiment Change in Twitter Streaming Data," *JMLR Workshop and Conference Proceddings 17:5-11*.

[77] Socher, R., J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning (2011) "Semi-supervised recursive autoencoders for predicting sentiment distributions," *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 151–161.

[78] Nguyen, L. T., P. Wu, W. Chan, W. Peng, and Y. Zhang (2012) "Predicting collective sentiment dynamics from time-series social media," *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*, pp. 6:1–6:8.

[79] Bae, Y. and H. Lee (2012) "Sentiment analysis of twitter audiences: Measuring the positive or negative influence of popular twitterers," *Journal of the American Society for Information Science and Technology*, **63**(12), pp. 2521–2535.

[80] Liu, K.-L., W.-J. Li, and M. Guo (2012) "Emoticon Smoothed Language Models for Twitter Sentiment Analysis," *In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*.

[81] Metaxas, P., E. Mustafaraj, and D. Gayo-Avello (2011) "How (Not) to Predict Elections," *Proc. of PASSAT/SocialCom*, pp. 165–171.

[82] Livne, A., M. P. Simmons, E. Adar, and L. A. Adamic (2011) "The Party Is Over Here: Structure and Content in the 2010 Election." *Proc. of the Fifth International AAAI Conference on Weblogs and Social Media*.

[83] Balasubramanyan, R., B. R. Routledge, and N. A. Smith (2010) "From Tweets to Polls : Linking Text Sentiment to Public Opinion Time Series," *Proc. of the Fourth International AAAI Conference on Weblogs and Social Media*.

[84] Tumasjan, A., T. O. Sprenger, P. G. Sandner, and I. M. Welpe (2010) "Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment," *Proc. of the Fourth International AAAI Conference on Weblogs and Social Media*.

[85] Mislove, A., S. Lehmann, Y.-Y. Ahn, J.-P. Onnela, and J. N. Rosenquist (2011) "Understanding the Demographics of Twitter Users," *Proc. of the Fifth International AAAI Conference on Weblogs and Social Media*.

[86] Mustafaraj, E., S. Finn, C. Whitlock, and P. T. Metaxas (2011) "Vocal Minority versus Silent Majority: Discovering the Opinions of the Long Tail," *Proc. of PASSAT/SocialCom*.

[87] Ceron, A., L. Curini, S. M. Iacus, and G. Porro (2012) *Every tweet counts? How sentiment analysis of social networks can improve our knowledge of citizens policy preferences. An application to Italy and France*, *Departmental Working Papers 2012-19*, Department of Economics, Management and Quantitative Methods at Universit degli Studi di Milano.

[88] Gayo-Avello, D. (2012) ""I Wanted to Predict Elections with Twitter and all I got was this Lousy Paper" A Balanced Survey on Election Prediction using Twitter Data," *CoRR*, **abs/1204.6441**.

[89] Skoric, M., N. Poor, P. Achananuparp, E.-P. Lim, and J. Jiang (2012) "Tweets and Votes: A Study of the 2011 Singapore General Election," *Proceedings of the 45th Hawaii International Conference on System Sciences*.

[90] Mejova, Y., P. Srinivasan, and B. Boynton (2013) "GOP primary season on twitter: "popular" political sentiment in social media," *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 517–526.

[91] Huang, S., M. Chen, B. Luo, and D. Lee (2012) "Predicting aggregate social activities using continuous-time stochastic process," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, pp. 982–991.