The Pennsylvania State University
The Graduate School
College of Information Sciences and Technology

EXPLOITING USER-GENERATED DATA FOR KNOWLEDGE DISCOVERY AND RECOMMENDATION

A Dissertation in
Information Sciences and Technology
by
Haibin Liu

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2014

The dissertation of Haibin Liu was reviewed and approved* by the following:

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

John Yen
University Professor of Information Sciences and Technology

Guoray Cai
Associate Professor of Information Sciences and Technology

Fuqing Zhang
Professor of Meteorology
E. Willard & Ruby S. Miller Faculty Fellow

David Hall
Professor of Information Sciences and Technology
Dean of the College of Information Sciences and Technology

*Signatures are on file in the Graduate School.

Abstract

The Internet and Web 2.0 have achieved a rapid growth and became ubiquitous in recent years. The advance in information technologies has also enabled users to generate data, implicitly or explicitly, on an unprecedented scale. Consequently, the need to discover and exploit new and useful knowledge from such data has also increased considerably. In this thesis, in this regard, we investigate user-generated data to discover interesting knowledge and enable better recommendation services.

First, we tackle the problem of the location type classification using individual Twitter messages. We extend probabilistic text classification models to incorporate temporal features and user history information as probabilistic priors, and show that the proposed models can boost the classification accuracy effectively.

Second, we study the problem of quantifying the notion of political legitimacy using collective Twitter messages for specific populaces. We design a framework that aggregates a large number of tweets into the final legitimacy score of a populace by leveraging probabilistic topic modeling and sentiment analysis technique. Our empirical evaluation on eight sample countries using related public tweets demonstrates that our proposed framework shows a strong correlation to results reported in political science literature. We also apply this framework to a traditional news media data set, and compare the results with Twitter data. Several interesting differences are discovered between these two medias for this quantification task of political legitimacy.

Third, we study the problem of mining implicit user feedback in recommendation systems. In particular, we tackle the cold-start problem of video recommendation using users' co-view information. We propose a classification framework to incorporate co-view information based on previously seen video pairs, and learn the weights of video attributes for ranking candidate videos to recommend, yielding encouraging recommendation results.

Finally, as a way to exploit social network for recommendation, we study the problem of recommending the best team for a given set of roles or skillset considering both individual and team characteristics. To quantitatively capture the team level features, we take various social networks among people into consideration from project history and many other online activities. Moreover, we learn the feature weights from the training dataset based on the correlation between features and project outcomes, and apply a combinatorial optimization algorithm to search the approximate best team. We validate our approach experimentally in a real business scenario and also compare our approach with other state-of-the-art methods using public DBLP dataset. The results demonstrate the effectiveness of our approach.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Internet and Web 2.0 have achieved a rapid growth and became ubiquitous in recent years. The advance in information technologies and emergence of a variety of online services have also assisted users to generate data, implicitly or explicitly, on unprecedented scale. Today people can publish and share various activities about their daily life online. They can write extensively in blogs like Wordpress [1], post pictures to Flickr [2], publish videos on Youtube [3], tweet through the microblogging application Twitter [4], and interact with friends via social networking services like Facebook [5]. As a result, the Web has become a tremendously rich repository of information with details about people's behaviors and activities.

Many online service providers have started leveraging such *user-generated data* a long time ago. Amazon[6] is such an example. To illustrate, Figure 1.1 shows an example product page of the camera *Canon 5D Mark II* on Amazon online store, along with various user-generated data on the same page. Figure 1.1a shows

[1] http://wordpress.com
[2] http://flickr.com
[3] http://youtube.com
[4] http://twitter.com
[5] http://facebook.com
[6] http://www.amazon.com

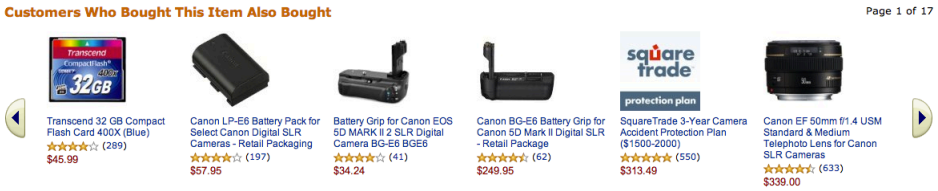
the product details of this camera, which are provided by the online service host. Figure 1.1b shows some other accessories users commonly bought together with the camera. Such patterns are usually extracted from users' purchase history. On the same page we can also see more detailed user ratings and reviews regarding this camera as shown in Figure 1.1c. Potential buyers can obtain much knowledge about the product details from such rating and review content. They can also get good accessory recommendations from the co-purchase patterns.

The aforementioned *user-generated data* portrays a wealth of resources full of individual, societal, and economical values. On the one hand, by analyzing this user-generated data, we can obtain information or understanding about people's habits, lifestyle, and thoughts etc. On the other hand, researchers and scholars can also rely on analysis over such data to gain in-depth understanding about the society or groups of people, from the collective perspective. Moreover, businesses value insights about users from such data so that for instance advertisements can be delivered more precisely. Consequently, studies about understanding and leveraging user-generated data have received extensive interests in recent years, both in industry and academic research [64].

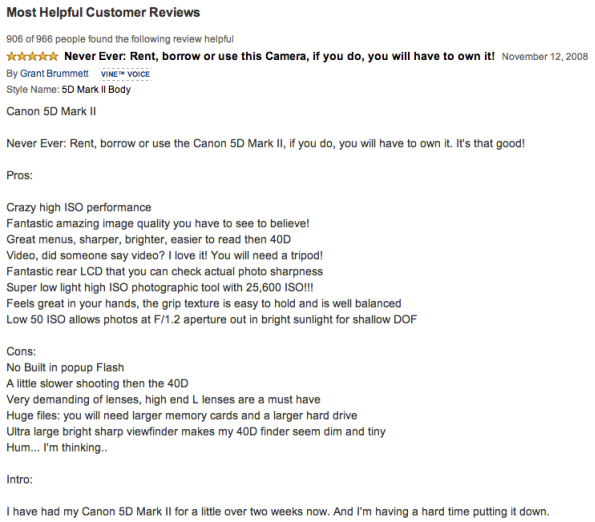
How can we effectively dig up and utilize the values inside such user-generated data? Our answer is through *data mining* [33]. Generally speaking, *data mining* means the process of discovering interesting and useful patterns, structures, and other valuable information from large amounts of data. The process of data mining is usually composed of five steps, as illustrated in Figure 1.2 [33]. After cleaning and preprocessing of the collected raw data, we carry out data mining algorithms over the transformed data to uncover the patterns and hidden structure. The results can be applied in a myriad of ways. However, they can be grouped into two general categories:

- **Discovering Knowledge**: Knowledge means interesting or useful information. However, there is no popular view or standard about determining the

(a) Product description on Amazon



(b) Co-purchase patterns from user purchase history



(c) User rating and review regarding the camera

Figure 1.1: A sample page of Amazon product with use of user-generated data

value of information. It must be user oriented or domain specific. With techniques of validation or visualization, the mining results can be interpreted by domain experts for their specific interests [28].

- **Developing Applications**: Applications like search engines and recommendation system can benefit greatly from the data mining results. For

example, we can apply classification technique to detect spam emails, or we can identify frequent purchased items from transaction history for product recommendation.

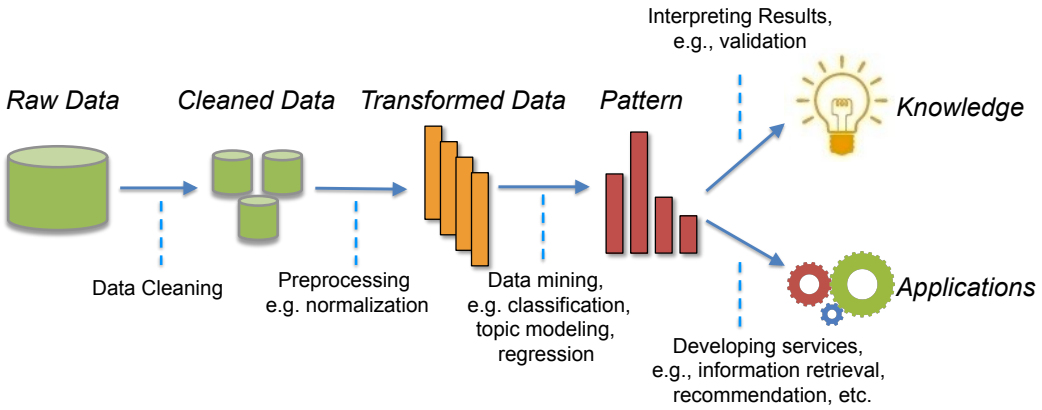


Figure 1.2: A Process of Data Mining

Although data mining techniques have been studied for decades, mining user-generated data is still quite interesting and challenging due to the following characteristics [64]:

- **Unstructured and noisy data.** The quality of user-generated data is not always guaranteed because they are mostly generated by amateur users voluntarily. The data mining technologies developed for ideal settings may not perform well when processing such user-generated data. For example, the language people everyday use is quite informal. As a result, content from social media is always full of spelling errors and grammatical mistakes. Abbreviations, slangs, and emoticons are also quite common. If we simply apply natural language processing technologies such as syntactic parsers that are traditionally developed over text of well written news articles or academic papers, the performance will be far from satisfactory. Instead, researchers may need to develop new techniques specially for such data.

- **Heterogeneous data.** Besides various data formats, recent developments of social networking systems introduce new social dimensions to the user-generated data. Studying interactions among users provides insights about users' interests. Incorporating social network influence with other information may also provide great benefits. For instance, Twitter may study users' habits from their Tweet messages, and and also learn influence of friends from social network, so that they can provide better personalized advertisements and recommendations.

In this dissertation, we investigate the problem of how to utilize data mining to improve understanding and leveraging user-generated data. We tackle the above challenges in mining social media data for knowledge discovery and exploiting user-generated data for recommendation services.

One typical and popular social media platform where users like to publicly publish their statuses and opinions is Twitter. A number of researchers have studied Twitter in order to obtain novel patterns or knowledge about users or the society [16, 26, 67, 69, 70, 74]. We add our pieces into these study efforts, and raise the first research question:

RQ1: ***What knowledge can we obtain from social media, and how do we achieve such knowledge?***

In the first part of this dissertation, we investigate two cases of short text mining over Twitter. We first tackle the problem of discovering location information from individual Twitter messages. More specifically, we try to infer users' location type from individual tweet content. Such location context in social media plays an important role because it is not only important in inferring social ties between people, but also vital for applications such as user profiling and targeted advertising. We utilize classification techniques to incorporate temporal and historical information to boost accuracy.

Then we study the problem of quantifying the notion of political legitimacy using collective Twitter messages for specific populaces. We design a framework with the advantage of topic modeling and sentiment analysis technique to achieve our goal.

Besides discovering knowledge to improve understanding, we are also interested in applying and leveraging such data and knowledge. Therefore our next research question is how to leverage user-generated data in applications and services. More specifically, we study one of the most widely applied services: the recommender systems. We study how such data can be used in practical recommendation scenarios,

RQ2: ***How do we utilize implicit data and social networks to improve real-world recommendation services?***

In the second part of this dissertation, we first study the problem of mining implicit user feedback in recommender systems. In particular, we tackle the problem of video recommendation using users' co-view information. We propose a classification framework to incorporate co-view information based on previously seen video pairs, and learn the weights of video attributes for ranking candidate videos to recommend.

Finally, as a way to exploit social networks for recommendation, we study the problem of recommending the best team for a given set of roles or skillset considering both individual and team characteristics. We take various social networks among people into consideration from project history and many other online activities. We learn the feature weights from the training dataset based on the correlation between features and project outcomes and apply a combinatorial optimization algorithm to search the approximately best team.

1.1 Contributions

We summarize the key contributions of this dissertation as follows:

- First we study location type classification problem using individual Twitter messages [54]. We extend probabilistic text classification models to incorporate temporal features and user history information in terms of probabilistic priors. The experiment results show that our extensions can increase classification accuracy from about 47% to 49% for overall dataset. However, in some specific daily time hours, the improvement is much more significant, e.g., from 37.7% to 45.3% for tweets posted at around 0 o'clock. We also propose a personalized location type classification model by incorporating users' check-in history. The experiment results demonstrate a boost in the accuracy from 47.1% to 57% for Maximum Entropy.

- Second, we then explore the problem of quantifying the notion of political legitimacy using collective Twitter messages for specific populaces. We design a framework that converts tweets into a number of topic dimensions using the probabilistic topic modeling, and leverage sentiment to evaluate the polarity of each tweet [53]. Our empirical evaluation on eight sample countries demonstrates that the proposed framework shows a strong correlation to results reported in political science literature [31], with the coefficient value of 0.7997887 (P-value = 0.01717). We also apply the same framework to a traditional news media data set, and compare the results with Twitter data. Several interesting differences are discovered between the traditional news media and social media for this quantification task of political legitimacy.

- Third, we investigate a problem of recommending new video lectures with the help of implicit user feedback. We propose a classification framework based on previously seen video pairs, and learn the weights of video attributes

for ranking candidate videos to recommend [52]. This framework leads to encouraging recommendation results. This framework leads to a mean average R-precision score of 25%, compared to the baseline of 21% without co-view information.

- Finally, we exploit social networks to recommend the best team for a given set of roles or skillset. To quantitatively capture the team level features, we take various social networks among people into consideration from project history and many other online activities. Moreover, we learn the feature weights from the training dataset based on the correlation between features and project outcomes. We apply a combinatorial optimization algorithm to search the approximately best team. We validate our approach experimentally in a real business scenario, and also compare our approach with other state-of-the-art methods using public DBLP dataset. The results demonstrate the effectiveness of our approach.

1.2 Structure of This Dissertation

The rest of this dissertation is organized as follows. The first part of this dissertation, Chapter 2 and Chapter 3, focuses on mining social media data, more specifically Twitter, to discover interesting knowledge. In Chapter 2 we explore our investigation about location type detection using tweet content. Chapter 3 presents our studies and discoveries about leveraging Twitter to quantify political legitimacy of countries. The second part of this dissertation, Chapter 4 and Chapter 5, focuses on exploiting user-generated data to improve recommendation services. In particular, Chapter 4 describes our work of mining implicit user feedback in a cold-start problem of video recommendation. Chapter 5 presents our team recommendation framework by exploiting social networks. Finally, Chapter 6 summarizes this dissertation and discusses the future research directions.

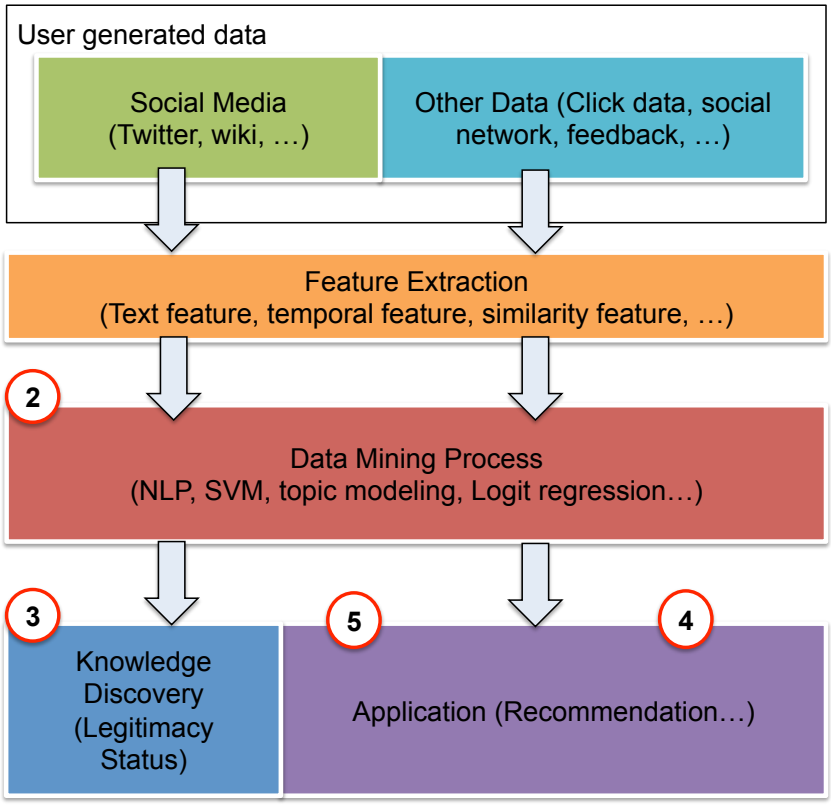


Figure 1.3: Different chapters in this dissertation fit into the problem of mining user-generated data with red circles indicating the chapter number.

We illustrate a general framework and workflow of leveraging user-generated data, as shown in Figure 1.3. The overall structure of this dissertation is also shown in the figure, with the red circles indicating the chapter numbers.

Chapter 2

Classifying Location Type Using Tweet Content

2.1 Introduction

Social media like Twitter has become a popular platform for people to share their daily activities and statuses. People can use location-based services like Foursquare, Google Latitude, Facebook Places, etc. to "check in" at venues and share them in social media. Besides, noncheck-in tweets in Twitter can also implicitly reveal their activity-level location context even if they do not explicitly publish it. On the one hand, activity-level location can reveal users' daily activities. We are interested in finding from tweets whether the user is working in office, dining in a restaurant, or exercising in a gym, etc. On the other hand, locations reveal further information with regard to people's behavior patterns and social interaction. For example, Figure 2.1 shows two sample tweets both of which talk about having dinner, however, their locations are different: the first one happens at home, the second is at a restaurant or a public event *Athletes Gala*. Taking such location context into consideration, we can infer that the first dinner is a pleasant gathering with a family member, while the second one is a fun hangout with friends.

Figure 2.1: Sample tweets that mention different activity locations

If we group users' location context into a few predefined types according to the characteristics of their activities, e.g., home is in the type *Residences*, and restaurants are in the type *Food* or *Nightlife Spots* according to Foursquare category list [1], we can obtain a clear understanding about involved user behaviors. Furthermore, if we can predict such activity level location types from users'public social media posts, we will not only arise users' privacy concerns, but also allow potential business service providers for targeted advertising.

In this chapter, we will study the problem of classifying location types based on content of users' check-in tweets. More formally, we define our research problem as:

Location Type Classification Given a stream of tweets $d \in D$, a fixed set of location types $C = \{c_1, c_2, \ldots, c_k\}$, and a training set S of tweets labeled with types of locations where they are posted $< d, c > \in D \times C$, we wish to learn a classifier γ that maps tweets to their context location types: $\gamma : D \rightarrow C$.

Many researchers have already studied the problem of revealing users' locations from tweets, and shown some promising progress [19]. Different from previous work that try to reveal city-level location from tweets, we are interested in the location context in a smaller scale, the activity level. This activity-level geographic information can be essential in many applications. To illustrate:

- Service providers can utilize activity level location information to present accurate targeted advertising.

[1] http://aboutfoursquare.com/foursquare-categories/

- The location type can potentially infer social ties between people. The presumption is that different social relationships have different interaction context. For example, as illustrated in Figure 2.1, if a twitter user tweets about activities regarding home, it is likely for her/him to interact with the family; if we find that the tweet location is a restaurant or a party, it is natural to infer that people will socialize with friends there.

- Location type revelation can also be used in user profiling. E.g., a user who tweets about Yellowstone National Park probably enjoys traveling, while a user who talks about beer in twitter is more likely to enjoy *nightlife*, or *food*.

- Studying location type detection can also arise people's privacy awareness. People who may not be willing to share their location information in non-check-in posts should be potential location can be detected from their post content.

Our goal is to filter out informative tweets the location type of each tweet using content only. More specifically, we will classify each tweet into one of the nine location categories listed by Foursquare. Our contributions in this paper include the following:

- First, we present in this paper a study of location type classification through a data set of informative location sharing tweets filtered from about 1 million check-ins.

- Second, we propose a probabilistic model to incorporate temporal features to improve classification accuracy. Accuracy by this model is improved slightly from about 47% to 49% for overall dataset. However, in some specific daily time hours, the improvement is much more significant, e.g., from 37.7% to 45.3% for tweets posted at around 0 o'clock.

- Third, we propose a personalized location type classification model by incorporating users' check-in history. The experiment results demonstrate a boost in the accuracy from 47.1% to 57% for Maximum Entropy.

The rest of this chapter is organized as follows: Section 2.3 describes baseline model and our proposed probabilistic models; Section 2.4 describes the process of our data collection and presents an analysis of data distribution; Section 2.5 shows our experiment results; we conclude this chapter in Section 2.6.

2.2 Related Work

Several researchers have investigated the problem of geo-location detection from tweet content [16,19,21,34,49]. Cheng et al. [19] tackle this problem in the city level. Purely based on the tweet content, the authors propose a probability language model to automatically identify words in tweets with a local geo-scope. [16] further improves user's home location prediction quality with Gaussian Mixture Models. They also employ an unsupervised measurements to rank the local words which remove the noises effectively. The authors in [49] are instead interested in the place of interest (POI) that a tweet refers to. The authors formalize the problem by ranking a set of candidate POIs using language and time models. Temporal factors need to be considered too because POIs are quite related to time. Because the POI related tweets are so sparse that the authors have to leverage search engine to enrich their language models. [21] addresses the geo-location detection problem in tweets from a different perspective. The authors are interested in matching a tweet to a specific restaurant. The question includes two parts: first, the authors need to detect which words mean a restaurant entity; second, if multiple restaurants have the same name, the authors need to detect which one is the exact match. In [34], Hecht et. al. study user behavior based on the location field in their Twitter profiles. They find that 34% of users do not provide real location information, but

frequently list fake locations instead. Nevertheless, they also find user tweets can help to infer the user's location with decent accuracy. But none of these studies are interested in the geo-location detection in activity scale like us.

Our research is directly related to the problem of location categorization. Two recent papers address this categorization problem [18, 92]. Researchers try to find out traffic patterns of venues from user generated check-in data, and take a further step to cluster the semantically related locations from these patterns. Traffic patterns can be defined as a vector of check-in frequency over a series of time units. For example, we can define daily traffic pattern that contains 24 time units, each of which represents an hour in a day; we can also define a weekly traffic pattern that contains 70 time units in which the time unit represents one tenth of a day [18]. [92] has a similar idea, in which the authors normalize the frequency into a probability density function, and call it temporal band. [18] shows that many categories indeed display quite similar daily temporal patterns, e.g., some coffee shops have similar high traffic in morning, and restaurants are frequently checked in at dinner time. [92] also demonstrates different geographic feature types have different weekly temporal bands. With such observations, the authors try to study further clustering and classification based on these similarities. But different from these papers, we do not have abundant features regarding each venue, nor are we interested in categorizing from venues' features. We instead focus on detecting location category from tweet content.

Our work is also related to short text classification. Several researchers tried to tackle this problem from different perspectives [20, 45, 72, 80, 83, 84]. Sriram et. al [80]study short text classification over tweets to help users better manage information from Twitter. Phan et. al [72] try to boost the classification accuracy by gaining external knowledge from Web search results. Notice that their classification is carried out over search snippets. Sun [83] tackles the short text classification task in an information retrieval framework. The predicted category is determined

by majority vote of the top search results. Researchers also examined whether general classification techniques can be carried out effectively over short text [10, 37, 56, 63, 73, 93–96]. [63] studied the problem of feature weighting in short text classification. In short text, each word or term occurs usually only once. In such situations, traditional tf-idf weighting strategy is not appropriate. Instead the author proposed a feature weighting approach called *Fragment Length Weighted Category Distribution*, and compared with TF-IDF, Chi-Squared, Mutual Information, and Information Gain. Liu et al. in [56] examined the feature selection strategy for short text. Proposed a feature selection method based on part-of- speech and HowNet. According to the composition of the text property, the authors choose the words with larger amount of information, and then expand the semantic features of these words based on HowNet, a knowledge base. Yuan et al. studied the smoothing methods of Naive Bayes in short text classification [93]. [73] studies combining latent semantic analysis (LSA) and independent component analysis (ICA) in short text classification. While LSA can be used to analyze and make use of co-occurrence of terms in text, ICA is good at classifying text with independent components of text documents. Combining these two may produce good results without generality in short text classification studies.

2.3 Location Type Classification

2.3.1 Baseline Methods

We aim to classify the check-in location types from tweet text content. Two commonly used text classification methods are Naive Bayes [57] and Maximum Entropy [66]. We first briefly introduce these baseline methods below.

2.3.1.1 Naive Bayes

If we look at each check-in tweet as a document d composed of a bag of words $w_1, w_2, \ldots, w_n$, we can compute the posterior probability that the check-in tweet belongs to category c as

$$\begin{aligned} p(c|d) &= \frac{p(c)p(d|c)}{p(d)} \\ &= \frac{p(c)p(w_1, w_2, \ldots, w_n|c)}{p(w_1, w_2, \ldots, w_n)} \\ &\propto p(c) \prod_{i=1}^{n} p(w_i|c). \end{aligned}$$

Note that $p(c)$ is the prior probability of a specific category, defined as

$$p(c) = \frac{N_c}{N}.$$

N_c is the number of check-in tweets in category c, and N is the total number of check-in tweets in training data set. The word distribution $p(w_i|c)$ can be estimated as

$$p(w_i|c) = \frac{N(w_i, c)}{\sum\limits_{w_j \in V} N(w_j, c)}$$

where $N(w_i, c)$ is number of occurrences of word w_i from category c. The check-in tweet is assigned to the best class determined by

$$\arg\max_{c \in C} p(c) \prod_{1 \leq k \leq n_d} p(w_k|c).$$

2.3.1.2 Maximum Entropy

Different from Naive Bayes, MaxEnt estimates the conditional probability directly in an exponential form instead of joint probability:

$$p(c|d) = \frac{1}{Z(d)} exp(\sum_i \lambda_i f_i(d, c)$$

where each $f_i(d, c)$ is a feature, λ_i is a constraint parameter to be estimated, and $Z(d)$ is the normalizing factor. In text classification, features are usually initiated as

$$f_{w,c'}(d, c) = \begin{cases} 0 & \text{if} c \neq c' \\ \frac{N(d,w)}{N(d)} & \text{Otherwise,} \end{cases}$$

where $N(d, w)$ is the number of times word w occurs in tweet d, and $N(D)$ is the number words in tweet d [66].

2.3.2 Location Type Classification: Our Proposals

We propose and explore two ideas to improve the accuracy of location type classification problem.

2.3.2.1 Temporal Model

In this subsection, we explore the impact of temporal features in the location type classification. Our assumption is that people prefer different activities at different time. For example, the location category of *Nightlife Spot* should be more frequently checked in at night than other time. Similarly, we expect more *Food* check-ins at meal times than early morning.

To leverage temporal impacts in our classification task, we divide all check-in tweets into 24 subgroups according to the hour of their posted time, and assign a new feature $t \in \{0, 1, \ldots, 23\}$ to every check-in. Now the classification problem

becomes

$$\arg\max_{c \in C} p(c|d,t)$$

which tries to maximize the conditional probability of a check-in tweet belonging to a location category given its content and posted time.

Hourly Prior Probability: One way to use this temporal feature is to apply hourly prior probability in text classifiers. Suppose the generative process of a user checking in a venue at a specific time is as follows: she first decides what kind of this check-in should be at current time, then she decides the content of that check-in tweet. Conditional independence is presumed here. That is, the content of the check-in tweet is determined only by the check-in category. More formally, in Naive Bayes, the joint probability becomes

$$p(c,d,t) = p(t)p(c|t)p(d|c).$$

For a given tweet, its posted time is always already known, the conditional probability can be estimated as

$$p(c|d,t) \propto p(c|t)p(d|c) \propto p(c|t)\prod_{i=1}^{n} p(w_i|c)$$

where $p(c|t)$ can be estimated as

$$p(c|t) = \frac{N_{ct}}{N_t}$$

while N_t is the number of check-in tweets in hour t, N_{ct} is the number of check-in tweets belonging to category c posted in hour t. $p(c|t)$ is called the hourly prior probability.

We also apply such hourly prior probability to Maximum Entropy classifier. However, since MaxEnt estimates the conditional probability $p(c|d)$ directly, as a

result, the hourly prior can be applied as

$$\begin{aligned} p(c|d,t) &\propto p(c|t)p(d|c) \\ &\propto p(c|t)\frac{p(c|d)p(d)}{p(c)} \\ &\propto \frac{p(c|t)p(c|d)}{p(c)}. \end{aligned}$$

2.3.2.2 Boosting with User Check-in History

Different users would apparently have different activity habits, therefore we would expect different personal check-in patterns accordingly. It is quite intuitive to guess that a student checks in more frequently at *College & University* than a white-collar worker. Therefore, simply applying a same overall prior probability for all users in tweet classification may not be fairly accurate for everyone. In this subsection, we discuss our exploration of incorporating users personal check-in history to boost classification performance .

Like hourly prior probability, we introduce a new user factor u in our model. Assuming independence between word distribution among categories and users' personal check-in habits, we can define the joint probability here as $p(c,d,u) = p(d|c)p(c|u)p(u)$, where $p(c|u)$ can be estimated from user u's personal check-in distribution. If we can retrieve adequate history check-in tweets for u, we can estimate

$$p(c|u) = \frac{N_{cu}}{N_u}$$

where N_{cu} is the number of history check-in from user u in category c, and N_u is the total number of history check-ins from u. As we are interested in maximizing conditional probability $p(c|d,u)$, and u is already known, the classification problem can formalized as

$$\arg\max_{c \in C} p(c|d,u)$$

and

$$p(c|d,u) \propto p(d|c)p(c|u).$$

Here the probability $p(c|d,u)$ includes two parts. While the first part $p(d|c)$ is a probability estimated from check-in tweet content, the second factor $p(c|u)$ is derived from user's personal check-in history. By replacing the category prior with personal check-in prior, we take both tweet content and personal habit into consideration.

Like hourly prior, to incorporate user check-in history into MaxEnt, the formula becomes

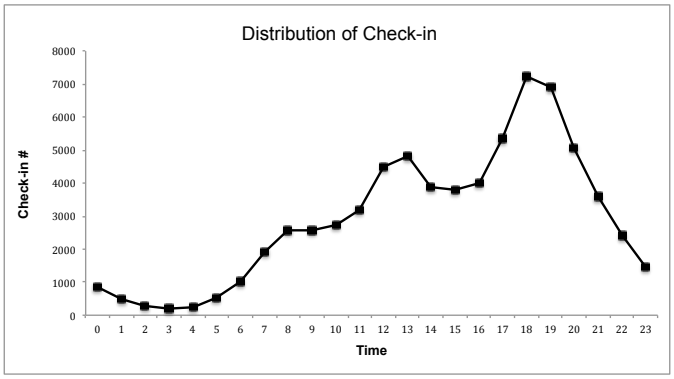
$$p(c|d,u) \propto \frac{p(c|d)p(c|u)}{p(c)}$$

where $p(c|d)$ can be estimated by MaxEnt, and $p(c)$ is the prior probability of category c in training data.

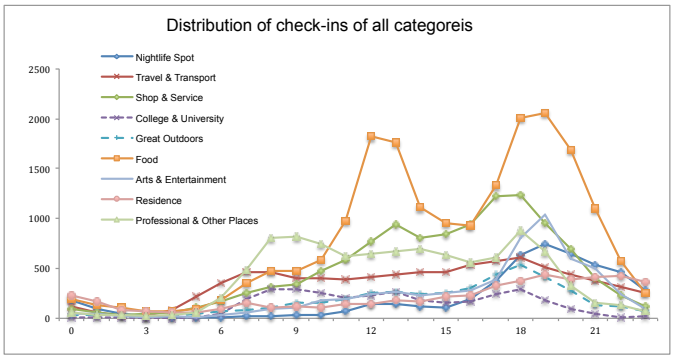
2.4 Experiment Setup

2.4.1 Data Collection

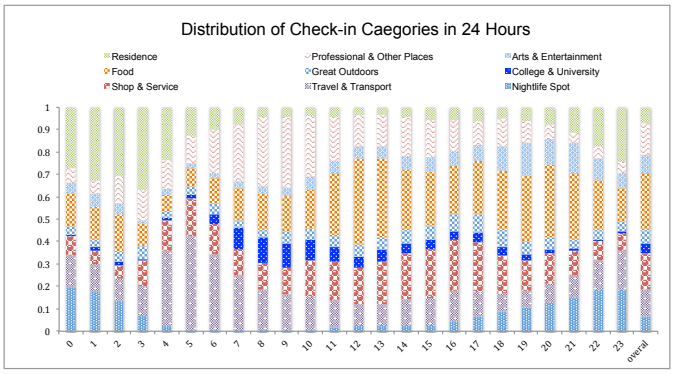
We adopt a data collection technique that relies on sampling Foursquare check-ins posted via Twiter. Using Twitter API, we search tweets with the keyword "4sq" because check-in tweets always contain URLs like "http://4sq.com/xxxxxx". We monitored Twitter's public streaming API and search API for a week in May 2012, and collected about 1 million tweets, among which there are more than 220,000 foursquare check-ins. Since our focus here is to classify context location types from tweet content, we removed check-ins that only contain venue information but no user-generated comments. We also filtered tweets with less than three words. Non-English tweets were also removed from our data set. Such filtering lead to a data set of about 120,000 check-ins. The foursquare URL embedded in each check-in tweet is linked to either a "venue page" or a "check-in" page, from



(a) # of check-ins in 24 hours



(b) Distribution of check-in for all categories



(c) Category distribution in 24 hours

Figure 2.2: Temporal distribution of check-ins

which we can retrieve more information about corresponding venues, and brief user profiles. Based on our best effort, we successfully tracked about 94,000 check-in tweets.

Foursquare has a hierarchy list of categories applied to venues, we use the top-level categories as ground truth to classify check-in tweets' location types. The

top-level categories are *Arts & Entertainment, College & University, Food, Great Outdoors, Nightlife Spot, Professional & Other Places, Residence, Shop & Service, Travel & Transport*. However, there are also some venues that are not assigned to any category yet, and some venues are labeled with more than one top category. We removed such data in our current experiments to simplify the setting. As a result, our experiment data set contains 72,643 check-in tweets with user-generated comments.

Table 2.1: Distribution of check-in tweet categories

Category	Percentage	# of check-ins
Arts& Entertainment	8%	5781
Travel & Transport	12%	8398
Professional & Other Places	14%	10006
College & University	4%	3206
Shop & Service	16%	11661
Nightlife Spot	7%	4916
Residence	7%	5089
Food	27%	19323
Great Outdoors	6%	4263

Table 2.1 shows the distribution of check-in tweet across categories in our data set. Among the nine categories, *Food* is the most popular one (27%); *Travel & Transport (12%), Professional & Other Places (14%)*, and *Shop & Service (16%)* are less popular; the other five categories have similar percentages (around 5%) in our data.

2.4.2 Temporal Feature

To explore temporal feature's impacts, we first need to retrieve temporal information of all tweets. Because we crawled tweets from all around world, it is necessary to convert check-ins' standard UTC into local time. Such localization requires timezone information from users. Although both Twitter and Foursquare provide posted or check-in time, we find that Foursquare covers more users than Twitter,

therefore we depend on Foursquare check-in API to extract tweets' localized post time.

Figure 2.2 shows the distribution of nine venue categories in our training data in 24 hourly time units of a day. Figure 2.2a demonstrates overall hourly check-in traffic pattern. Each point in this plot shows the number of check-in posted in an hour in our data. It shows that people check-in most frequently at 18 or 19 o'clock during a day. Figure 2.2b further illustrates detail distribution for each category. This shows us the check-in traffic changes along hours for every category. For example, we can see that *Food* are more frequently checked in at around 12 and 19 o'clock than other time of a day, and 19 is the most frequently checked in hour for *Nightlife Spot* venues. Figure 2.2c demonstrates category distribution in each hour. We also append the overall category distribution to this plot. This helps us understand not only the difference of distribution among hours, but also between each hour and the overall percentage. Compared to hours from 13-20, the category distributions in early hours like 0-8 are quite different from overall distribution. This plot also shows us which categories are the most dominant in each hour. We can see that although Table 2.1 shows that *Food* is the overall dominant category, this is not always the case in individual hours. For example, at 5 AM *Travel & Transport* venues are quite more frequently checked in than *Food*, also *Professional & Other Places* venues are more popular than any other category at 9 AM.

2.4.3 User Check-in History

To evaluate the boosting impact of user check-in history in classification, we collected another data set by crawling the latest up to 1,000 tweets from randomly selected 252 users. Each of them have at least 30 check-in tweets. The total number of check-in from these users is 50,929. Distribution of the user check-in number is shown in Figure 2.3.

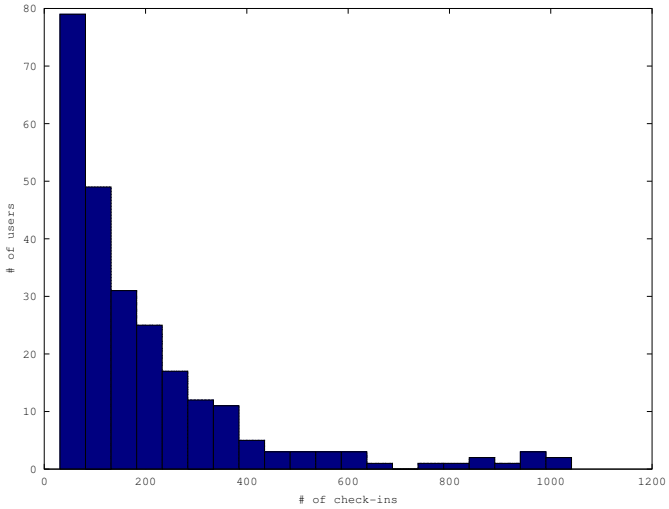


Figure 2.3: Distribution of user history check-in

2.5 Experiment Results

The experiments are performed using Mallet toolkit [60]. Extensions are also implemented over Mallet package. The performance of all classifiers is compared in the measure of accuracy across all classes, calculated as

$$\text{accuracy} = \frac{\text{number of true positives}}{\text{number of test data set}}.$$

We use stringent five folds cross validation, and the final results are averaged over the five folds.

Figure 2.4 reports the results of baseline methods of Naive Bayes and MaxEnt, and also our extension of temporal model. Applying hourly prior improves overall accuracy from 47.3% to 48.6% in Naive Bayes, and from 48.6% to 49.8% for MaxEnt. We also report the details of classification performance for data in each hour in Figure 2.5. It shows that HourlyPrior+NB and HourlyPrior+MaxEnt achieved significant improvement in most of the hours, especially in the early hours of 0-10. This can be explained by the difference between category distribution in

Table 2.2: Top Features of Each Venue Category in Location Classification in MaxEnt

Category	**Top 15 Informative Feature Tokens**
Arts & Entertainment	aveng, battleship, mib, dictat, fantasm, shadow, expect, movi, lampion, sox, preview, museum, yanke, siff, globaltv
Travel & Transport	bound, plane, layov, hotel, flight, tsa, airport, bali, termin, land, airplan, cancun, buse, jakarta, runway
Professional & Other Places	adjust, mri, mail, checkup, bibl, scholarship, worship, permit, dentist, mass, marchayosoy, choir, pastor, juri, patient
College & University	lectur, campus, exam, class, assign, student, workshop, account, studi, clase, semest, quiz, ceremoni, lab, librari
Shop & Service	haircut, pedicur, slurpe, yoga, ab, stock, pedi, shop, gas, bicep, trim, treadmil, store, tricep, cardio
Nightlife Spot	pub, pint, cider, whiskey, adag, sproutup, afterparti, trivia, beer, drink, karaok, dart, deserv, patio, bar
Residence	nighti, goodnight, apart, cuddl, homey, bed, rest, sleep, throne, balconi, bath, bedtim, shower, dormir, home
Food	meal, latt, brunch, sushi, comer, caffein, pho, dwolla, coffe, lunchi, margarita, burrito, espresso
Great Outdoors	hike, wharf, picnic, jog, golf, trail, kickbal, basebal, beach, sail, bench, softbal, swim, leagu, magnific

these hours and overall distribution as shown in Figure 2.2c. Because the early hours' distribution is more different from overall distribution than other hours, the improvement is also accordingly higher by applying specific hourly prior in these hours. We also note that during the hours like 11-15, all methods have similar performance. The similar category distribution patterns during these hours with overall category distribution could also explain such classification resemblance.

Results of incorporating user history check-in are demonstrated in Figure 2.6.

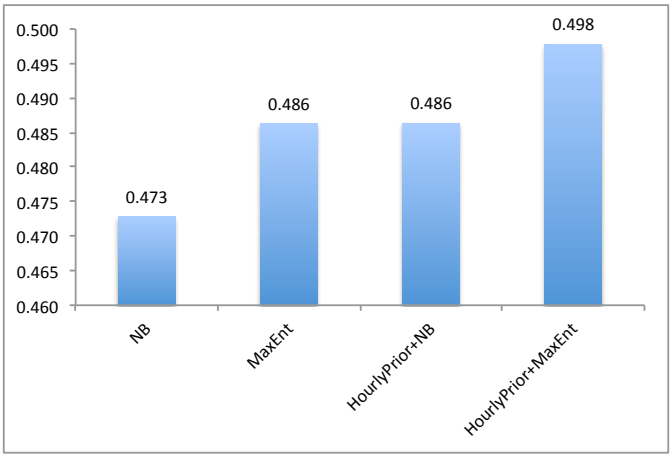


Figure 2.4: Overall classification accuracy

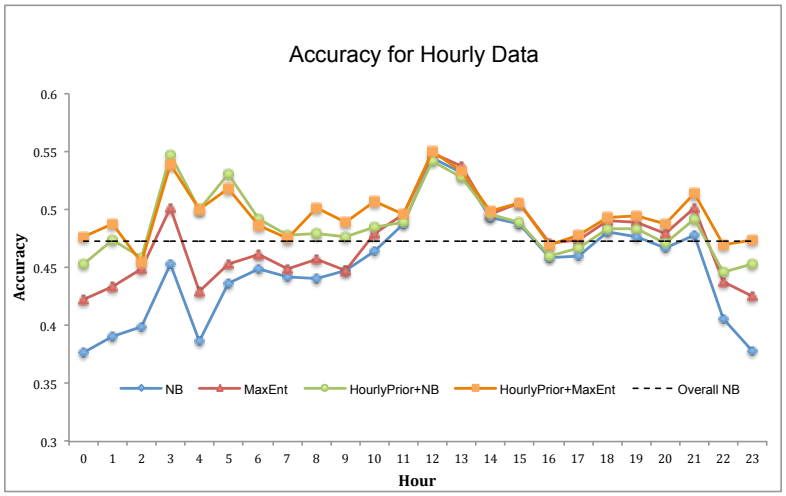


Figure 2.5: Classification accuracy for each hour's data

It shows that MaxEnt+UserHistory has the highest accuracy, 57.0% , compared to original 47.1% in this data set. We notice that 41.5% accuracy can be achieved using history distribution only. That is because many users are quite apt to specific venue categories. Some users may simply repeat checking in exactly the same venues. However, when they check in venues different from the history dominant category, we have to rely on tweet content for prediction.

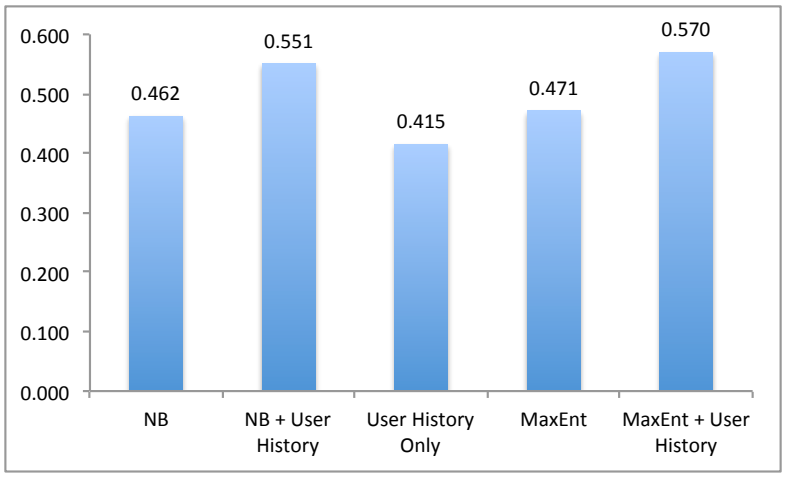


Figure 2.6: Classification accuracy with help of user check-in history

2.6 Conclusions

In this chapter, we study the problem of classifying location types based on content of users' check-in tweets. We extend basic classification models by incorporating temporal features and user behavior history. The experimental results show temporal features can achieve decent performance improvement, especially in hours when the data distribution is quite different from overall daily distribution. Personal check-in history also effectively boosts the classification performance significantly.

Our future work will focus on further improving classification accuracy. One common problem in classifying short text like tweets is data sparseness. To address such sparseness problem, we will study feature selection and augmentation techniques with regard to location types. Another direction is to investigate social factors. Since people interact with their friends and followers in various locations, it will be interesting to integrate social network data in this location type classification problem. We will crawl check-in data from users' friends and followers, and study the correlations between their check-in patterns. Strategies to integrate such social data into our classification framework need to be carefully studied in future.

Chapter 3

Quantifying Political Legitimacy Using Tweets

3.1 Introduction

The term *political legitimacy* in political science refers to the acceptance of authority by a law, government, or civil system, and has been the subject of extensive study in the discipline. The concept is often viewed as "central to virtually all of political science because it pertains to how power may be used in ways that citizens consciously accept" [30]. As such, in political science, many proposals have been made to quantify the legitimacy of a populace. Some recent works such as [30, 31] have been well received in the community. While useful, however, such existing works are largely based on hand-picked small-size data from governments or UN based on an ad hoc formula. Therefore, it is still challenging to renew or expand the results from [30, 31] to other regions if there exist no reliable base data.

To address this limitation, in this research, we ask a research question "*if it is possible to quantify political legitimacy of a populace from social media data*", especially using Twitter data. As a wealth of large-scale public tweets are available for virtually all populaces, if such a quantification is plausible, the application

can be limitless. For instance, in the stochastic simulation environment such as NOEM [77], a quantified legitimacy score forms one of important input parameters. While there is currently no good way to synthetically generate a legitimacy score of a populace, one may be able to estimate it from the tweets generated from or closely related to the populace.

3.2 Related Work

3.2.1 Quantifying Political Legitimacy

Our idea is mainly inspired by Gilley's work [30, 31] from political science. Gilley tries to measure political legitimacy of 72 countries around the world using collected survey data such as World Value Survey, Global Barometer regional surveys etc. The author defines legitimacy of a state as follows: *"a state is more legitimate the more that it is treated by its citizens as rightfully holding and exercising political power"*. Legitimacy quantify is used in terms of degrees as a continuous variable. He also proposes to weight views of all citizens equally in measuring legitimacy.

In order to measure the latent state legitimacy meaningfully, Gilley tries to quantify three constitutive sub-variables [30]:

- *Views of legality* refers to how legally citizens think the state has acquired and exercises political power about laws, rules and customs.
- *Views of justification* means citizen responses to the moral reasons given by the state for the way it holds and exercises power. It focuses on the rightfulness.
- *Acts of consent* means positive *actions* that express a citizen's recognition of the state's right to hold political authority and an acceptance to be bound to obey the decisions that result.

Moreover, Gilley examines available data sets throughly and selects several quantitative indicators to measure the aforementioned variables. These indicators can be found in the data for most interested countries, and thus can be measured accordingly for most countries. For example, the attitude surveys about corruption, views of police, judges and civil servants are used for the measurement of views of legality; views of effectiveness of political institutions, popularity of embedded polity are used to measure views of justification; election turnout, voter registration, military recruitment are used to quantify acts of consent.

Furthermore, how to aggregate all these variables is the next question. The author proposes several different ad-hoc weighting strategies that values various aspects differently. One way is to take all indicators equally and use an unweighted manner, another is to prioritize the views of justification more than the other two variables. The final reports show quite close results for these two strategies.

We can see that the whole process is quite dependent on available data sets. On the one hand, the author has to select data that are commonly available for most countries. On the other hand, the data collection process per se is quite time consuming, leading to a long time before a replicate of measurement can be updated [31].

3.2.2 Mining Social Media

In recent years the exploitation of social media such as Twitter and Facebook to predict latent patterns, trends, or parameters has been extensively investigated. For instance, [81] computationally tried to classify tweets into a set of generic classes such as news, events, or private messages. In addition, [27, 46, 69] attempted to track and analyze the status of public health via social media data. Some even tried to predict stock market from public mood states collected from Twitter [12]. Studies have also been carried out about the correlation between tweets' political sentiment and parties and politicians' political positions [85, 86]. The case study

Figure 3.1: Tweets related to legitimacy.

about 2009 German federal election [85] reported a valid correspondence between tweets' sentiment and voters' political preference. Such studies also verify that the content of tweets plausibly reflects the political landscape of a state or region. Another paper [68] also aggregates text sentiment from tweets to measure public opinions.

While closely related, our method focuses on quantifying the political legitimacy, that is related to not only politics and elections, but also other concepts such as governments, laws, human rights, democracy, civil rights, justice systems, etc. To our best knowledge, this is the first attempt to computationally quantify the political legitimacy of a populace from a large amount of big social media data and conduct a correlation analysis against the results in political science.

3.3 The Proposed Method

Our goal is to build and validate a model to accurately quantify the political legitimacy score of a populace for a specified time period using tweet messages. The underlying assumption is that some fraction of populace would occasionally express their opinions on the status of political legitimacy. Two such examples are shown in Figure 3.1. The first mentions about the democratic situation of Egypt, while the second expresses the concern about the justice system in the US.

Let us use the term **L-score** to refer to the political legitimacy score of a

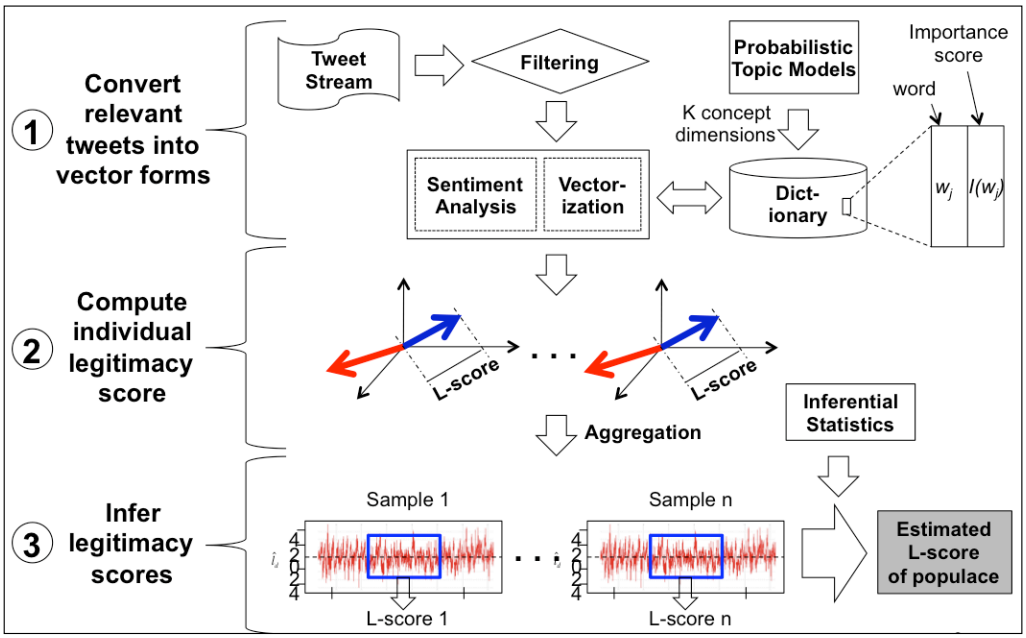


Figure 3.2: Overview of the proposed method.

populace, scaled to a range of $[0, 10]$. Then, our overall method consists of three steps: (1) identify and convert relevant tweets into computable feature space, (2) compute L-score of each tweet, and (3) aggregate L-scores to form a time series and compute final L-score of a populace. This overall workflow is illustrated in Figure 3.2.

3.3.1 Step 1: Vectorizing Tweets

Each tweet can be up to 140 characters but often very terse. The challenge of this step is to be able to accurately capture and extract critical features from short tweets that can indicate the opinion of a writer toward the status of legitimacy. Since there is no widely-accepted “computable" definition of legitimacy, we assume that the notion of political legitimacy is related to k-dimensional topics such as justice system, human rights, democracy, government, etc. While treating k as a tunable parameter in experiments, then, we simply attempt to represent each

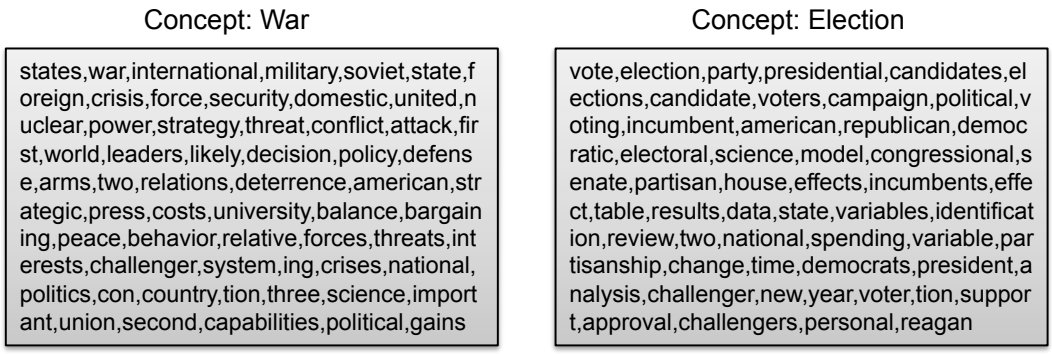


Figure 3.3: Two prominent topics found from political science journal articles.

tweet as a k-dimensional vector, where the score in each dimension indicates the relevance of the tweet to the corresponding topic. Further, we use a dictionary of k dimension where each dimension (i.e., topic) contains a set of keywords belonging to the topic. Finally, we run a probabilistic topic modeling technique such as Latent Dirichlet Allocation (LDA) [9] over politically oriented corpus[1] and build such a k-dimensional dictionary.

Figure 3.3 illustrates two example topics found by LDA and prominent keywords within each topic (the labels such as "war" and "election" are manually assigned). Note that, although found automatically, such topics represent the main themes of the corpus reasonably well and can be viewed as related to the legitimacy. In addition, prominent keywords within each topic also make sense. Therefore, if a tweet mentions many keywords found in either topic, then the tweet is used to quantify the legitimacy.

Suppose k topics are first manually selected and corresponding keywords in each topic are found using LDA. Imagine a k-dimensional dictionary such that a membership of a keyword can be quickly checked. For instance, one can check if the keyword "military" exists in the "war" dimension of the dictionary. Furthermore, suppose each keyword, w, in the dictionary is assigned an importance score, $I(w)$. In practice, a tf-idf [91] style frequency-based score or LDA-computed probability score

[1] http://topics.cs.princeton.edu/polisci-review/

can be used to measure the importance of keywords. For instance, an importance of a word can be computed using the following frequency-based formula:

$$I(w) = \frac{freq(w)}{\sqrt{1 + (freq(w))^2}}.$$

Using this data structure of the k-dimensional dictionary, we can convert tweets into vectors and then compute the L-score.

With such a topic dictionary, we can convert each tweet into a k-dimensional vector by checking membership of words in each dimension. Assume that a tweet, t, is pre-processed using conventional natural language processing (NLP) [40] techniques such as stemming and represented as a bag-of-words, w, with n words: $t \Rightarrow w = \{w_1, w_2, \cdots, w_n\}$. Then, the k-dimensional vector representation of a tweet, v_t, is:

$$v_t \in R^k = [\alpha_1 \sum\nolimits_{\forall m_1 \in |w \cap D_1|} I(m_1), \cdots, \alpha_k \sum\nolimits_{\forall m_k \in |w \cap D_k|} I(m_k)]$$

such that $\sum_1^k \alpha_i = 1$, D_i refers to the i-th dimension of the dictionary, and α_i is the weighting parameter for the relative importance of the i-th dimension.

3.3.2 Step 2: Computing L-scores of Tweets

The intuition to compute L-score of a tweet is that when a tweet either positively or negatively mentions keywords related to k-dimensions of the legitimacy, their "strength" can be interpreted as the legitimacy score. The L-score of the tweet, $L - score(t)$, is then defined as the magnitude (i.e., L2-norm) of v_t, with the sign guided by the sentiment of the tweet t–Δ_{sent}. Suppose $v_t = (x_1, ..., x_k)$. Then,

$$L - score(v_t) = \Delta_{sent} \|v_t\| = \Delta_{sent} \sqrt{x_1^2 + \cdots + x_k^2},$$

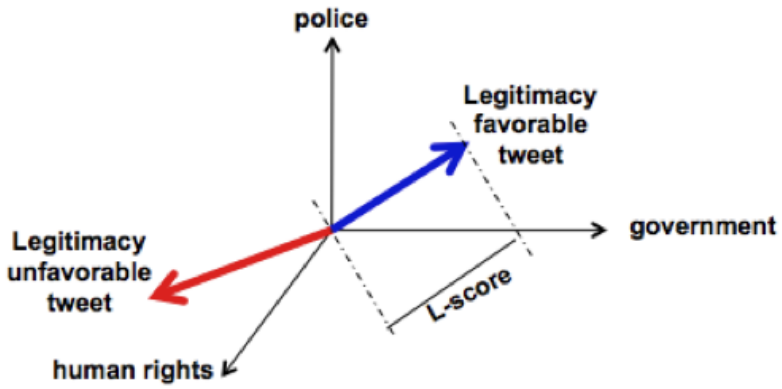


Figure 3.4: Illustration of L-score in vector space.

where Δ_{sent} indicates a $[-1, 1]$ range of sentiment polarity score of the tweet. Note that an alternative to this single Δ_{sent} per tweet is to allow for different sentiment polarity per dimension, Δ_i , in each tweet. Therefore, the vector representation of a tweet has the following formula, sightly different from the above one:

$$v_t \in R^k = [\Delta_1 \alpha_1 \sum\nolimits_{\forall m_1 \in |w \cap D_1|} I(m_1), \cdots, \Delta_k \alpha_k \sum\nolimits_{\forall m_k \in |w \cap D_k|} I(m_k)]$$

such that $\sum_1^k \alpha_i = 1$.

While this alternative is a more general formula as it allows each dimension to have a different sentiment (and thus polarity score of [-1, 1]), in our preliminary study, as typical tweets are rather short and there are usually simply not enough information to determine different polarity score per dimension, we maintain a single sentiment score per tweet.

Figure 3.4 illustrates the process of the computation of the L-score, visually, using 3 arbitrary concept dimensions (i.e., K=3, police, government, human rights). We show the process with a tweet example in the following.

Example 1 L-Score Computation *Consider the example tweet in Figure 3.5. After the pre-processing, the tweet can be represented as the following bag-of-words: t ={"corruption", "administrator", "more", "disgusted", "system", "see", "get",*

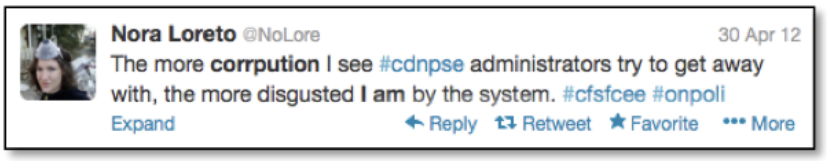


Figure 3.5: Example tweet for L-score computation

"#cdnpse", "#cfsfcee", "#onpoli",⋯ }. Further, suppose we use $K = 4$*{gov, justice, police, human-rights} in the dictionary.*

- *gov:* $D1 = \{government : 0.9, administrator : 0.8, \dots\}$
- *justice:* $D2 = \{system : 0.7, court : 0.9, corruption : 0.3, \cdots\}$
- *police:* $D3 = \{police : 0.9, corruption : 0.5, \cdots\}$
- *human-rights:* $D4 = \{brutality : 0.7, right : 0.5, \cdots\}$

Then, the vector representation of the tweet can be as follows:

$$\begin{aligned} v_t &= [1/4(I(\text{“}administrator''), 1/4(I(\text{“}system'') + I(\text{“}corruption'')), \\ &\quad 1/4(I(\text{“}corruption'')), 0] \\ &= [0.2, 0.25, 0.125, 0]. \end{aligned}$$

Note that the word "corruption" is used twiceâĂŤonce in the "justice" and second in the "police" dimension dictionary. Same word can belong to multiple dimensions in the dictionary (along with different important scores).

3.3.3 Step 3: Aggregating L-scores of Tweets

Once the L-score has been computed for all tweets, we next need to aggregate all the L-scores per some "group" and determine the representative L-score of the group. One example grouping constraint can be a region (e.g., country such as Egypt or city such as Detroit). Suppose we want to aggregate all L-scores of the day

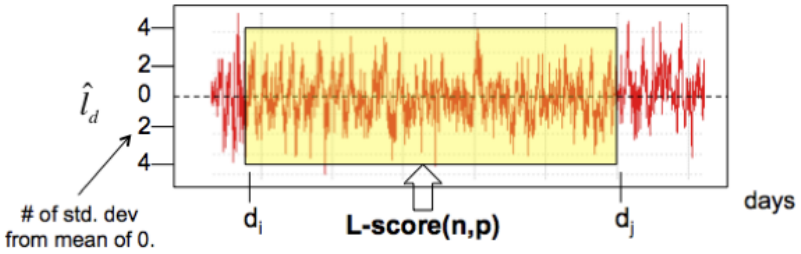


Figure 3.6: Example time series of L-scores.

d. Assuming the distribution of the daily L-scores follow the Gaussian Distribution, then, we first compute the mean L-score of the day as,

$$\hat{l}_d = \frac{\sum_{\forall l_i \in L_d} l_i}{|T_d|}.$$

Then, the Z-score normalization, similar to [11], can be applied using the mean L-score as follows:

$$\tilde{l}_d = \frac{\hat{l}_d - \mu_{pop}}{\sigma_{pop}},$$

where μ_{pop} and σ_{pop} indicates the mean and standard deviation of entire population. Since such statistics of the entire population is not available, instead, we use the interval-based estimated Z-score normalization such as the following:

$$\tilde{l}_d = \frac{\hat{l}_d - \mu_{d,\pm\delta}}{\sigma_{d,\pm\delta}}, with[d-\delta, d+\delta],$$

where both mean and standard deviation are estimated from a specific time window, instead of the whole population.

When daily representative L-scores are computed for a specific period, $p : [d_i, \cdots, d_j]$, we have a time series as follows:

$$\tilde{l}_{TS}[d_i, d_j] = [\tilde{l}_{d_i}, \tilde{l}_{d_{i+1}}, \cdots, \tilde{l}_{d_j}].$$

Such a time series is illustrated in Figure 3.6.

Table 3.1: L-scores published in [31].

Country	Score	Country	Score	Country	Score
Norway	7.97	Japan	6.13	India	5.21
Bulgaria	3.21	Canada	7.26	Thailand	5.89
Vietnam	7.07	United States	5.83	Brazil	4.68
New Zealand	6.78	South Africa	5.45	Slovenia	4.33
Iran	2.04	France	5.03	Peru	3.44
Spain	6.64	China	5.36	Turkey	3.96

Once a time series is created for the specified group, we can employ standard time series analysis techniques to either compute the overall representative score of the entire time series, or predict future L-scores. For instance, in the current implementation, we used both moving average (MA) [3] and auto-regressive MA (ARMA) models [14].

3.4 Empirical Validation

3.4.1 Evaluation Metric

Since there is no ground truth to L-scores of populaces, as an alternative, we aim to see "if our method yields L-scores of populaces similar to those reported in [31]." For instance, Table 3.1 shows example L-scores reported in [31]. This, computed from UN and WHO data, is widely accepted in political science community.

3.4.2 Tweet Data Collection

We chose eight countries with varying L-scores in [31]–i.e., Brazil, Iran, China, Japan, Norway, Spain, Turkey, and USA. We prepared two sets of data: (1) *Geo* dataset contains tweets generated within the bounding box of the geo-coordinates of each country of interest, and (2) *Keyword* dataset contains tweets that mention terms related to each country (e.g., a hash tag of "#USA"), regardless of their geo-coordiates. Since the Geo-dataset are always from within the boundry of interested

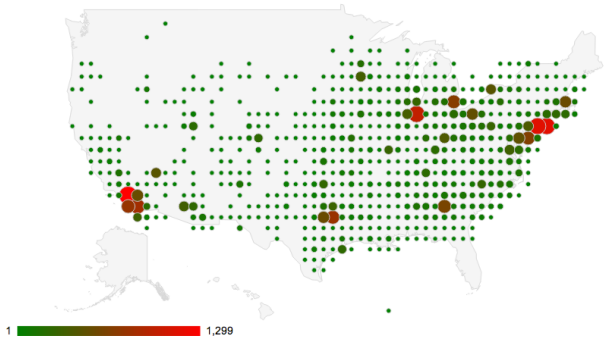
Figure 3.7: Examples of tweets about "#Brazil"

countires, we can guarantee the relevance of crawled tweets in this set. For the Keyword-dataset, we have the hash tag of country name to indicate close relevance to each country. Figure 3.7 shows two sample tweets having the "#Brazil" hash tag, and quite related Brazil's political legitimacy status. As a result, the Geo-dataset can be deemed as *internal* attitude towards the political legitimacy of specified countries, while the Keyword-dataset can be considered *external* perspective about the political legitimacy of the countries.

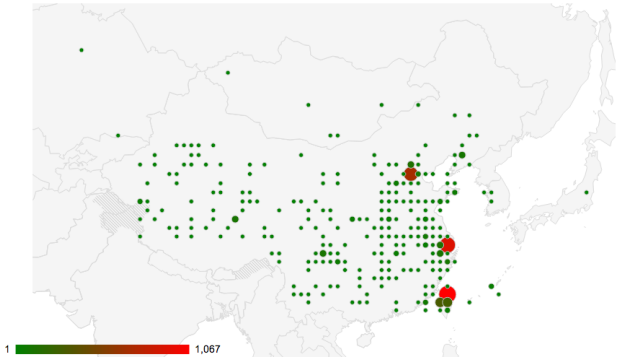
From 9/28/2013 to 11/6/2013, we collected a total of 300,450 tweets using Twitter streaming API[2] that are written in English. The two data sets are monitored and crawled using location and keyword constraint separately. Only relatively meaningful tweets were retained in our experiments (e.g., terse tweets with less than 4 words or location-based tweets having the form of "I'm at location" are removed). Table 3.2 summarizes statistics of tweets that we used in the experiments. The third column in the table (# Filtered Geo Tweets) refers to the tweets after the aforementioned filtering process.

Figure 3.8, for instance, shows the geo-coordinates of tweets in the Geo dataset for USA and China. We also overlay collected tweets from all eight countries based on their geo-coordinates on top of one world map. The distribution illustrates the coordinate constraints are quite accurate for each country. Interestingly, we also observe that there are a few "hubs" that have more tweets than most other

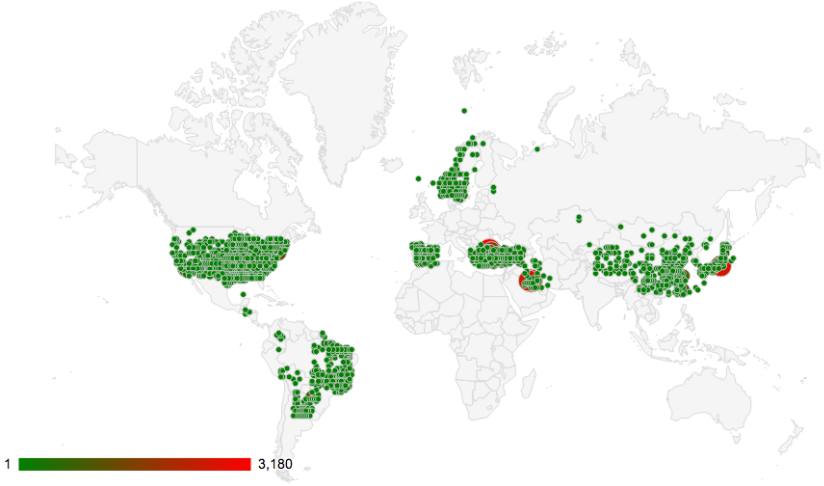
[2]https://dev.twitter.com/docs/api/streaming



(a) USA



(b) China



(c) All Countries

Figure 3.8: Geo-cordinates of tweets in Geo datasets.

areas in each country. For example, in the USA map, we see tweets from the few metropolitan areas, such as New York City, Chicago, have larger red circles, compared to the smaller green circles in many other areas. This is also in accordance

with the population distribution in the country.

Table 3.2: Summary of crawled tweets.

	# Keyword tweets	# Geo tweets	# Filtered Geo Tweets
Brazil	10,924	18,788	14,715
China	17,848	8,060	7,569
Iran	51,743	9,600	6,594
Japan	13,112	9,948	9,427
Norway	6,561	5,633	5,554
Spain	15,845	13,094	12,477
Turkey	13,281	38,187	14,634
USA	28,801	39,025	38,662

3.4.3 Sentiment Verification

One of the key factors in our framework is the sentiment analysis. In this subsection, we present our inspection about the sentiment analysis results. We use the Pattern.en sentiment analysis engine to analyze sentiment over tweets [3]. Pattern is a web mining module for Python, and the Pattern.en module is its natural language processing (NLP) toolkit [24]. It scores sentiment based on the English adjectives used in the text of a sentence with the help of WordNet. The toolkit will output a polarity value ranged between -1.0 and +1.0. We treat polarity value larger than 0 as positive, and less than 0 as negative.

To verify the sentiment toolkit's effectiveness, we randomly selected 362 tweets from our data set. The tweets are nearly equally selected from the eight countries and two data sets, we removed tweets that are too short (less than five words). We first manually scrutinized these tweets and assigned sentiment labels to these tweets as ground truth. Then we run the sentiment analysis toolkit and compare the results with manually assigned labels. In our experiments we only leveraged positive and negative tweets. In this verification data set, 61.9% are positive, and 38.1% tweets are negative.

[3] http://www.clips.ua.ac.be/pages/pattern-en

The overall accuracy of the toolkit over this verification data set is 77.1%. This accuracy is actually quite high compared to many other sentiment analysis experiments over social media in an automatic machine learning manner [5, 32, 76, 78, 89]. It is reported that it is hard to achieve sentiment accuracies larger than 80% for binary positive/negative classification for single sentences [78]. For the more difficult multiclass case including a neutral class, accuracy is often less than 60% for Twitter [89]. Our sentiment analysis results are on par with these reports. Some other tool like Stanford sentiment tool [78] is reported to have higher accuracy, yet our experiments over our sample data set actually do not achieve better results (with accuracy of less than 60%). We have seen one work with extremely high precision of 99% over Twitter messages [36]. However, that work is rule-based and involves extensive human work to build rules. It is not easy to re-build same model without long time manual labeling work.

Ideally, in order to achieve the accurate sentiment for tweets, we should recruit human raters to check all instances of tweets (e.g., carried out on the platform of Amazon Mechanical Turk). However, that would take much time and cost to renew every experiment. Since our sentiment results are acceptable compared to state-of-art studies, we would stick to that tool for our current L-Score experiments. We would like to employee human efforts to improve the accuracy in our future experiments, either by training a better analysis model or fully depend on human efforts.

3.4.4 L-Score Results

In this subsection we present the aggregated mean L-score of crawled tweets during the monitored period. Several factors are studied that may affect the final L-score. First, the number of topics obtained from LDA may play an important role in quantifying tweets' score. We tried different number of topics from 4 to 20, and the results are shown in Figure 3.9, where $Dict4$ means result from dictionary with

4 topics. Note that the range of the L-scores are rescaled to $[0, 10]$ to be compliant with the results of [31]. We can see that different number of topics lead to slightly different L-scores on both Geo and Keyword datasets.

It is worth to put a few more notes about the topic dictionary here. First, LDA groups words according to their co-appearance frequency, therefore it does not guarantee that every topic would be semantically meaningful according to our political study needs. We notice that we may be able to identify some prominent topics like 3.3, yet there are many other topics that cannot be easily assigned to any pragmatic theme. For example, since the topic modeling is carried out over political academic corpus, some words like "Tocqueville", "Platos" are quite related to political theory or history, which may not be so common in everyday conversation in tweets. Second, while the first few topics vary substantially, there are also some overlaps in some topics. For example, the word "political" almost appears in every topic, and the word "government" appears in 9 out of 20 topics. This leads to the idea that the first few topics may differentiate tweets more significantly than the other topics. Third, since Twitter messages are usually quite short, many messages are even less than 20 words. Therefore it may be more pragmatic to keep the dictionary dimension small than to utilize the complete 20 topics. Table 3.3 shows some words in the first eight topics.

Studies are also carried out to see the impact of granularity of sentiment analysis in calculating L-scores. While previous results are calculated using sentiment polarity scaled in range $[-1, 1]$, we also tested with only extreme sentiment values of $\{-1, 1\}$. However, the L-scores using this extreme sentiment values show little difference.

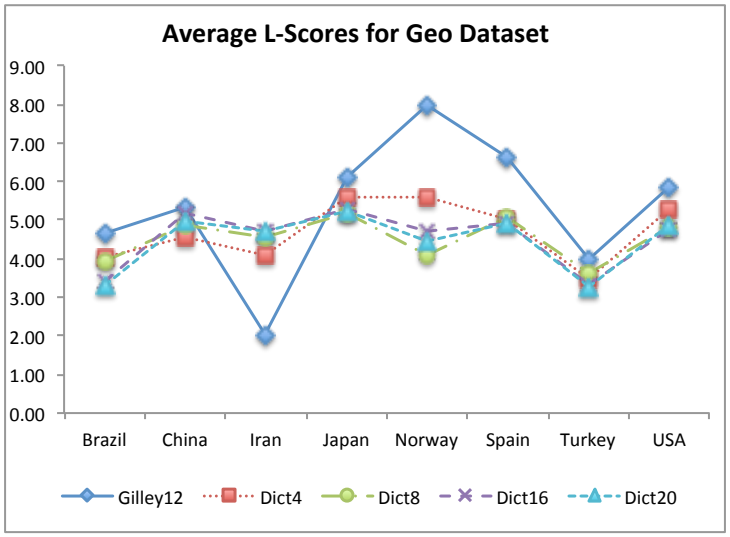
As mentioned previously, we use the scores reported in [31] as a proxy to "ground truth" in this experiment, and examine how correlated our result is to the score in [31]. To see the overall correlation with [31], we computed the *Pearson correlation coefficient* (PCC) [75] between the L-scores of all of our methods (using different

Table 3.3: Words in the first eight topics.

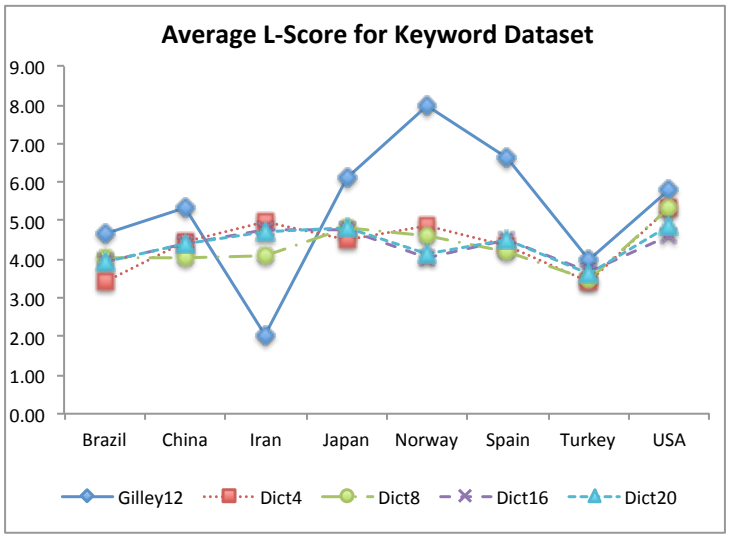
political, social, economic, labor, power, politics, century, revolution, union, movement, industrial, communist, struggle, unions, radical, democracy, elite, revolutionary, tocqueville, regime, reform
social, politics, approach, empirical, behavior, power, rules, march, rational, evidence, assumptions, issues, development, logical, role, validity, rationality, events, context, freedom
war, international, military, soviet, crisis, security, domestic, nuclear, threat, attack, leaders, defense, peace, forces, threats, crises, weapons, escalation, adversary, stability
vote, election, candidate, campaign, political, republican, democratic, congressional, senate, partisan, president, approval, reagan, impact, republicans, campaigns, races
equilibrium, probability, utility, distribution, proposition, uncertainty, payoff, risk, assumptions, strictly, proof, bias, assumed, resources, informed, legislators, benefits, marginal
collective, individuals, members, rational, benefits, dilemma, protest, incentives, prisoners, payoffs, economic, interest, payoff, participation, provision, repeated, institutions
political, moral, justice, freedom, aristotle, virtue, women, ethics, marx, community, liberty, plato, individual, law, morality, ethical, rational, activity, capacity, truth, equality, desire
income, economic, tax, government, social, population, labor, capital, trade, market, benefits, economy, federal, investment, distribution, employment, expenditures, governments, welfare, debt, capita, redistribution, wage

number of topics or sentiment values) and [31]. As shown in Table 3.4, the best performer is the $Dict4$ over Geo dataset. With the coefficient value of 0.7997887 (P-value = 0.01717), we can claim a better correlation between L-score computed using $Dict4$ and Geo dataset and that reported in [31] than other configurations.

We also examined details of the tweet data set, and found some interesting discoveries that merit brief comment here. First, the two data sets crawled in different ways show quite different patterns. We saw people express their opinions or comments regarding political issues in both data sets. For example, in Geo data set, we could see that people would express their opinion about the government



(a) L-scores of Geo dataset



(b) L-scores of Keyword dataset

Figure 3.9: Aggregated mean L-scores.

or politician, like: "*Sooo can just the ppl who voted for Obama take Obamacare? I mean seriously.. If you were that dumb. cause you obviously don't understand it.*" There are similar tweets in Keyword data set, like the following one with hashtag of #Norway: "*In #Norway there are more men that commit #suicide than who dies in traffic and murder combined. Governments ignores the issue #mensrights.*" However, while Geo data set contains mostly people's daily life

Table 3.4: PCC values between L-scores of our proposed methods and [31].

	Keyword	Geo	Keyword-Extreme	Geo-Extreme
Dict4	0.2035	**0.7998**	0.2142	-0.4527
Dict8	0.4729	0.2334	0.4015	-0.5389
Dict16	-0.0634	0.3756	0.2709	-0.5943
Dict20	0.0705	0.3071	0.1880	-0.6310

activities or comments, a large proportion of Keyword data set contains links that redirect to other online news media content. We can tell whether a tweet is disseminating online news media content from that if it contains URL that directs to other sources. The proportion of tweets with URLs in two data sets are quite different. For example, 63.2% tweets from Keyword data set for Iran have URLs, while only 26.5% in Geo data set contain URLs. This pattern indicates a close relation between social media and other online news media.

Second, we also found that while Geo data set is quite scattered in daily topics, Keyword data set has some prominent topics or issues that are frequently mentioned, usually in the form of retweet which means people simply repost some messages from other users. We examined some message details for Turkey and Brazil, and found during the crawling time, the topics of "Syria", "student protest", "redhack", and "millionmaskmarch" appeared quite frequently in Turkey Keyword dataset. About 10% of filtered tweets with non-zero sentiment using Dict4 were talking about Syria issues with the hashtag of Turkey. About 67% of such tweets are detected negative sentiment. 5% of tweets were about university student protest against Turkey's higher education. We noticed that around 11/06/2013, when our crawling was being carried out, students of METU organized one protest with wide support across the country, and also evoked some riots[4]. And about 4% were retweet of a message of supporting "MillionMaskMarch" and "RedHack", which Was initiated by a group called "Anonymous" to protest around the world [5]. For Brazil, one

[4] http://en.wikipedia.org/wiki/2013-14_protests_in_Turkey

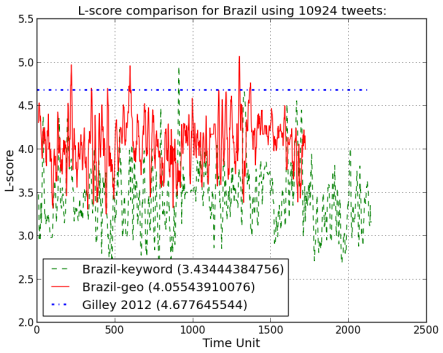
[5] http://en.wikipedia.org/wiki/Anonymous_(group)

frequent topic is world cup. About 5.7% in our filtered data set mentioned that topic. Actually many tweets (almost 50%) were criticizing Brazil for holding the event. One frequently retweeted message is: *"What if the money invested in the World Cup were spent on Brazilian education?"*

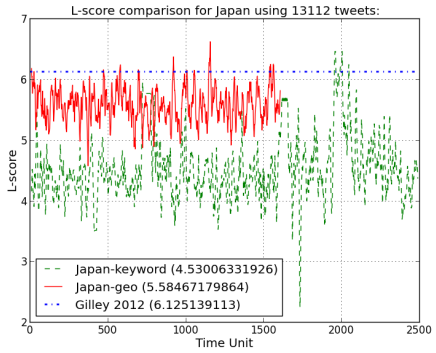
We expect the political status of a region or country would stay relatively the same for a short period, thus we represent the status of a country with one average score. However, the above observation indicates that Twitter is actually quite related to timely events, and therefore the L-score achieved from tweet messages would also display frequent fluctuations. It would be also interesting to look at time series of L-score on a daily basis. Such studies and results will be presented in later section in this chapter.

There are also some pitfalls in our method of data collection. Using only hashtags may not accurately target the interested countries. One tweet we found in China's keyword data set is: *"#Zambia has issued arrest warrants for three international players who missed the 2-0 friendly defeat by Brazil in #China"*. This message used a hashtag of China but here China simply works as the place where the game was held, not the real involved subject of the interested issue. However, the message was retweeted more than 200 times, almost 5% of the tweets regarding China after filtering. We have to admit that such noise is somehow inevitable, but still most of the data are still directly related to our study subject.

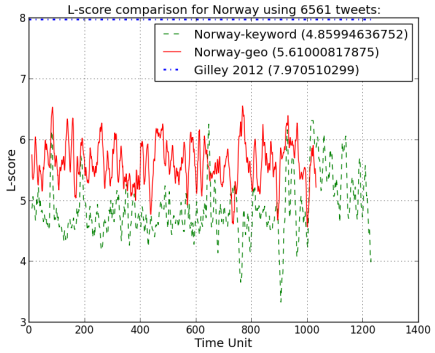
Figure 3.10 shows time-series of 4 countries using 4 LDA topics on Geo and Keyword datasets. In most cases, L-scores estimated from Geo tweets match better than those estimated from keyword tweets. Note that compared to L-score of [31], our estimation of L-score matches well for some countries but poor for others (e.g., Norway).



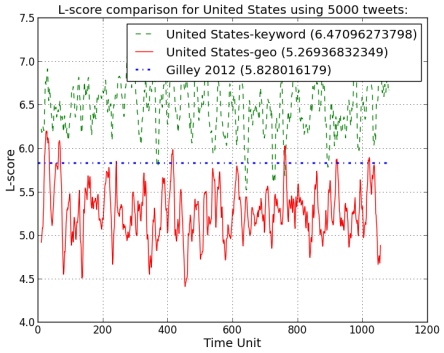
(a) Brazil



(b) Japan



(c) Norway



(d) USA

Figure 3.10: L-score time series of 4 countries with 4 LDA topics on Tweet dataset.

3.5 Further Experiments Using GDELT Data

In previous section, we report the comparison and correlation between legitimacy scores calculated from tweets and the results from political academic report. In this section, we further compare the legitimacy scores using two different media sources: traditional news media and social media. Traditional news media usually covers political events with many details, while social media express broad concerns from a mass of common people. There may be common and also different characteristics when leveraging them for the purpose of quantifying legitimacy. Therefore, we carry out the comparison study and report our discoveries in this section.

3.5.1 Data collection and experiment setup

We use the Global Database of Events, Language, and Tone (GDELT) database as the representative of tradition news media [6]. GDELT is a data set that tries to brief global political events with coded geo-location and tone [48]. The data are based on news reports from a variety of international news sources. We believe that this data set contains information useful for legitimate status of the covered countries.

We collected two data sets for the eight countries: Brazil, China, Iran, Japan, Norway, Spain, Turkey, and USA, for a time period during 4/25/2014 and 5/15/2014. One captures the traditional media from GDELT database, and the other does the social media from Twitter. For GDELT data, the web site [7] maintains a database of Global Knowledge Graph (GKG) that captures the persons, organizations, themes, events, and tones that occur in global news. GKG provides global daily updates since April 1, 2013 to present. Since GDELT data are mainly collected from English news agents, they can be deemed as an external attitude towards the target country's legitimacy status. For the same purpose, we leverage Twitter stream API [8] with hash tags of all the eight countries (e.g., use "#usa" to crawl tweets related to USA) to collect only English tweets as external attitude towards the countries from social media. Table 3.5 summarizes the number of news records and tweets we crawled for our experiments.

GDLET does not store source text of news. However, instead it automatically detects and assigns to each news a list of themes. For example, a news article "Ukraine protesters confront police anew after nation's bloodies day" [9] is assigned themes including "TERROR", "PROTEST", "KILL", "LEGISLATION", etc. The

[6] `http://gdeltproject.org`

[7] `http://data.gdeltproject.org/gkg/index.html`

[8] `https://dev.twitter.com/docs/api/streaming`

[9] https://my.news.yahoo.com/ukraine-police-charge-protesters-nation-39-bloodiest-day-004816528.html

Table 3.5: Summary of collected data.

	# of news records	# of tweets
Brazil	15,455	151,042
China	101,643	168,355
Iran	28,592	416,414
Japan	47,980	137,892
Norway	9,414	39,970
Spain	25,463	115,497
Turkey	26,121	155,271
USA	767,012	818,064

catalog of themes are predefined according to some coding rules. There is a description of each theme in the catalog, e.g., the theme of "KILL" is described as "Any mention of something dying". We replace all themes with their according description sentences, and apply our quantification framework over this description text to detect the relatedness of each news record to the political legitimacy purpose.

On the other hand, GDELT also calculates the tone value of each news. More specifically, it first calculates positive score as the percentage of words in the article that were found to have a positive emotional connotation, and negative score as the percentage of words with negative emotional connotation. The final tone value is calculated as positive score minus negative score. We treat this tone as sentiment value in our quantification framework. Due to the calculation method of the tone value, potential range for tone value is [-100, 100]. However, a simple examination tells us that most tone values lie in the range of [-10, 10]. Therefore, we divide the range into seven sections and count the number of records whose tone value lies in the corresponding bin: <-20, [-20, -10), [-10, -5), [-5, 5], (5, 10], (10, 20], >20. As shown in Figure 3.11, about 80% of the records have tone value in the range of [-5, 5], and 98% of the records lie in [-10, 10]. To make the scenario simple, we curtail

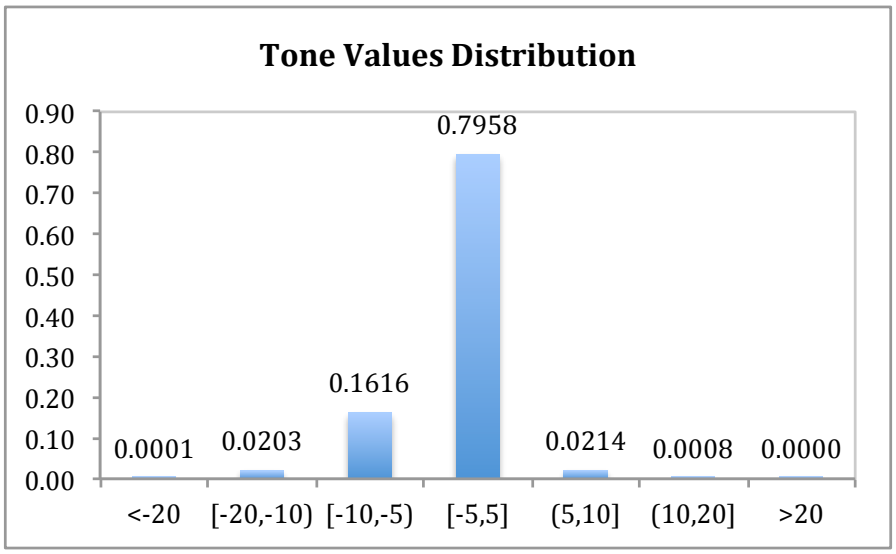


Figure 3.11: Distribution of Tone Values in GDELT Data

the tone value range to [-10, 10] by applying a segmentation function as follows:

$$\text{tone} = \begin{cases} -10 & \text{if tone} < -10 \\ tone & \text{if } -10 \leq \text{tone} \leq 10 \\ 10 & \text{if tone} > 10 \end{cases}$$

For the experiments, since the different topics of dictionary lead to similar results, we only report our results using 4 topics dictionary.

3.5.2 Verification of GDELT data

Since GDELT dataset collects political news from traditional media, we expect it to be an effective indicator about political legitimacy around the world. To verify its efficacy, we carry out a case study about the country Ukraine using data in GDELT web site collected from 08/01/2013 to 05/15/2014. We calculate the daily L-score based on our framework of Chapter 3 by leveraging the tone value and event themes annotated in GDELT, apply smoothing average over a window of seven days, and normalize the final score. The result is shown in Figure 3.12. As a

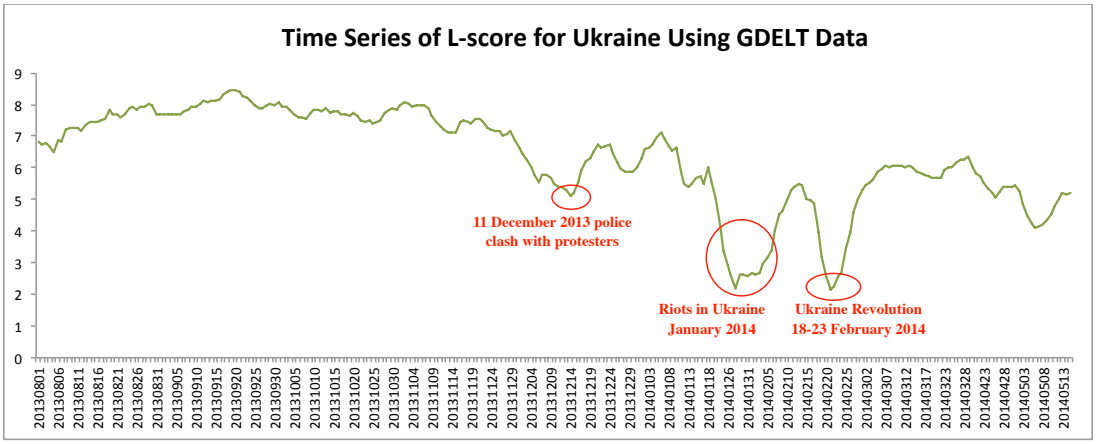


Figure 3.12: Time Series of Daily L-score for Ukraine from 08/01/2013 to 05/15/2014

wave of demonstrations and public protests demanding closer European integration happened in Ukraine in late 2013 (called Euromaiden [10]), and a revolution took place in February 2014 after a series of violent events (2014 Ukraine Revolution [11]), there was considerable civil unrest in Ukraine during that time. Accordingly, we would expect dramatic drop of L-score during the end of 2013 and the beginning of 2014. We check the major events from the timeline on Wikipedia, and annotate these events that happened at according dates in the time series of L-score in Figure 3.12. The figure shows L-score has corresponding fluctuation for these major events. This conforms to our aforementioned assumption.

3.5.3 Quantifying legitimacy for countries

We calculate daily average L-score for all countries, and obtain the smoothed average using a range of four days. The results are plotted in Figure 3.13 for GDELT data set and Figure 3.14 for Twitter data, respectively. The results demonstrate that L-score derived from traditional media differs more greatly among countries than social media data does. As shown in Figure 3.13, L-score derived from traditional

[10] http://en.wikipedia.org/wiki/Euromaidan
[11] http://en.wikipedia.org/wiki/2014_Ukrainian_revolution

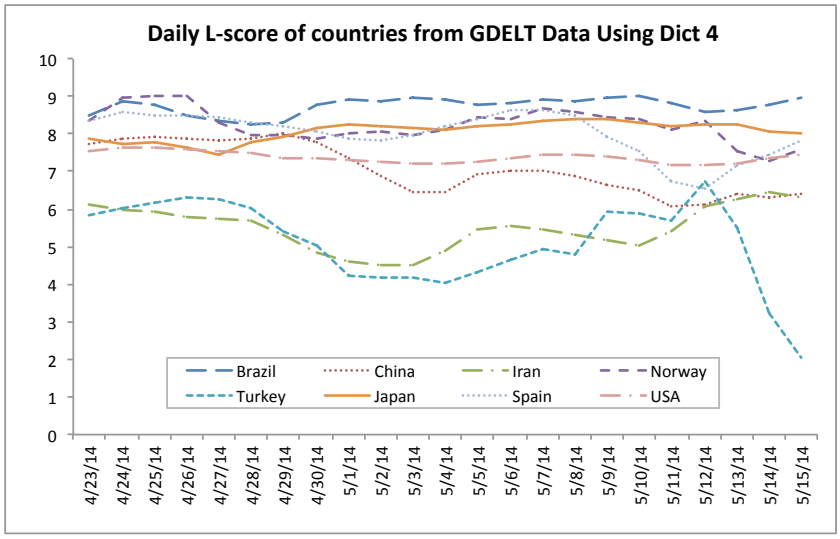


Figure 3.13: Daily L-score of GDELT Dataset

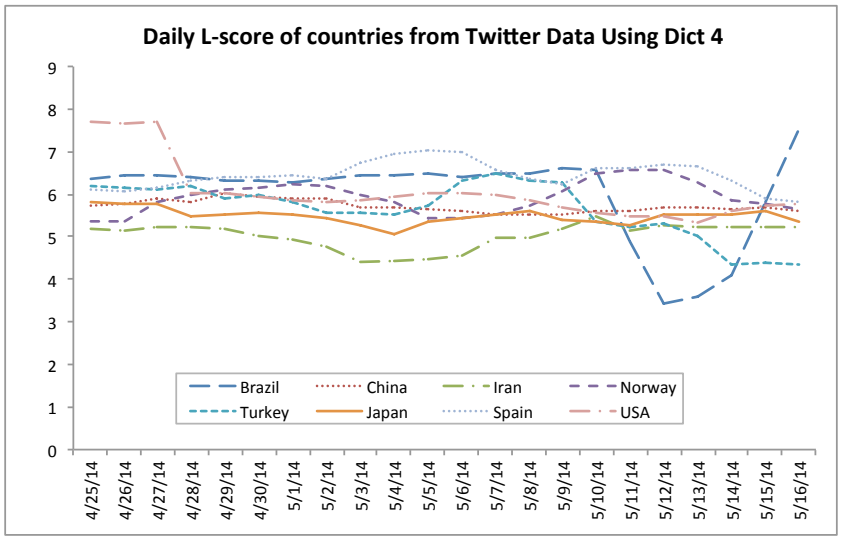


Figure 3.14: Daily L-score of Twitter Dataset

media for countries Iran and Turkey during this time period are mostly lower than that derived from other countries. However, in Twitter data, while Iran is almost always the lowest, Turkey does not show the same inferiority compared to other countries.

Due to the normalization process we applied to the two data sets, although we try to rescale the L-score of both data sets to the range of [0,10], it may still be

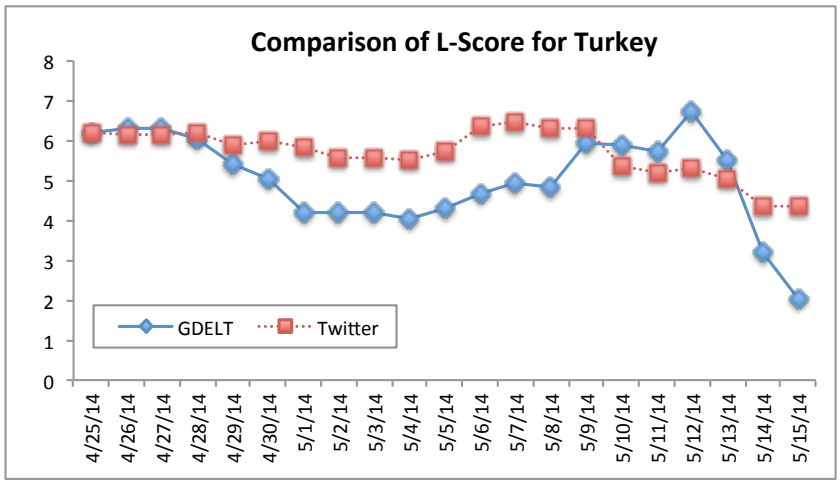


Figure 3.15: Comparison of Daily L-score Distribution for Turkey

inappropriate to compare the values directly between them. Instead, we look at the score fluctuation trend and relative ranking of the studied countries.

We study two cases for the country of Turkey and Brazil. As shown in Figure 3.15, the L-score derived from both GDELT data set and Twitter data set for Turkey show similar fluctuation during this time period. However, GDELT data set shows a more dramatic drop of the score at around 5/14/2014 than Twitter data does. As we scrutinize the data around the dates between 5/13/2014 and 5/15/2014, and check the major events that happened during that time, we believe that the major cause for this L-score drop is the happening of *Soma mine disaster* on May 13, 2014 [12], which killed 301 people. Both traditional media and social media talked a lot about this event, yet we find that the difference is mainly caused by the coverage proportion. For example, on May 14, 2014, 1,900 out of 2,508 (75.8%) news articles in the traditional media collected by GDELT were talking about the killing. However, in social media of Twitter, 2,022 out of 8,399 (24.1%) politically related tweets were directly pertinent to the Soma accident. From this observation, we believe that social media sometimes are more diversified about the social or political topics than the traditional media.

[12] `http://en.wikipedia.org/wiki/Soma_mine_disaster`

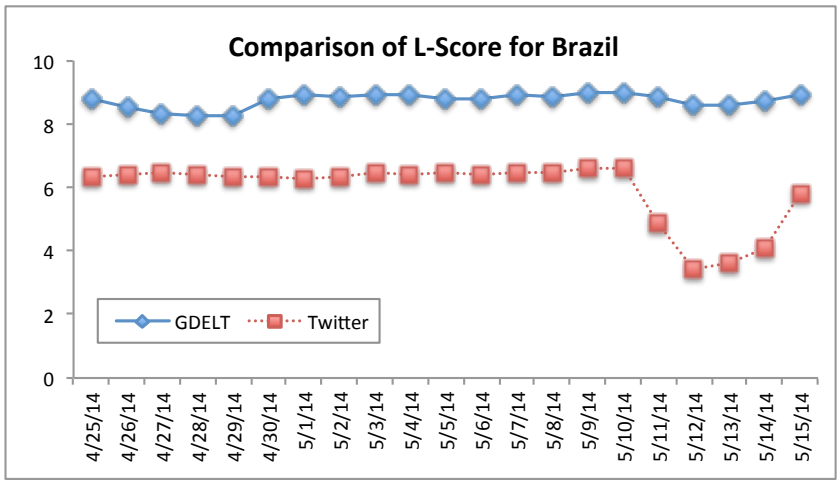


Figure 3.16: Comparison of Daily L-score Distribution for Brazil

On the other hand, as shown in Figure 3.16, we also notice an obvious drop of L-score for Brazil on May 12, 2014 in Twitter data set, while the GDELT data does not show much change (notice that as we explained previously, the L-score difference between two data sets may come from the normalization process). We find that on May 12, 2014, there were 3,346 of retweets of the same message that talks about "The Brasilian government is evicting people from poor areas to make room for the world cup". On that day, in addition, a total of 4,909 tweets were politically related. This causes the dramatic drop of L-score using Twitter data on the day. However, the message was originally published on May 1, 2014 actually, but have an tremendous impact via retweeting on May 12, 2014. This may also explain why we cannot find related news from traditional media in GDELT data set on the same day. In this case, the social media may not work as a news media, but rather a tool for a social movement. This functional difference from traditional media also leads to different results in terms of people's opinion towards the legitimacy status.

3.6 Conclusions

In this chapter, we study the problem of quantifying political legitimacy of a populace based on public Twitter data. We propose a solution that converts short tweet text messages into a number of topic dimensions using probabilistic topic modeling. Then we leverage sentiment analysis to evaluate polarity of each tweet, and aggregate a large number of tweets into the final legitimacy score of a populace. Our experiments over real tweets collected about eight countries reveal that some configuration of our proposal shows a strong correlation to results reported in political science community. We also carry out similar quantification framework over traditional news media, and compare the results with Twitter keyword data set. We discover several interesting differences between these two different media.

Despite the promising result, there are a set of *limitations* to our study:

1. In general, our framework points a good direction towards our legitimacy quantification goal. However, there are still small drawbacks in our method that needs further improvement. The sentiment results may be improved if we can leverage human labler to examine carefully. We would also improve the classification model by providing a large mannually labeled training twitter data set. The dictionary can also be improved if we can consult domain experts so that we can get tweets that are accurately related to political issues.

2. To derive a more definite conclusion on the validity of our proposed method in quantifying the legitimacy, a more comprehensive experiment is needed– e.g., more number of countries, larger tweet datasets, or topics derived from different corpus;

3. While [31] is a reasonable "beta" ground truth for our study, there is no formal analysis why or how accurate it is. As such, more correlation analysis

of our proposal using different methods to compute the legitimacy is needed;

4. Although we also carry out the comparison between the legitimacy scores using news media like GDELT and social media of Twitter, the study needs further verification. For example, we may need further information about the news records to detect their more accurate relation to political legitimacy; we may also need to crawl larger tweet datasets with longer time so that a more comprehensive study can be carried out.

Chapter 4

Exploiting Co-view Information for Lecture Video Recommendation

4.1 Introduction

Recommender system is a typical research field that heavily exploits user generated data, especially implicit user feedback. Given a set of users and a set of items, the goal of a recommender system is to predict the items a particular user is most likely to be interested in. Recommending products for users on a shopping website like Amazon, predicting the ratings that a user is likely to assign to a movie, predicting the citations a paper is likely to make are some common scenarios where automatic recommender systems are desirable [6, 51, 82]. One of the most widely applied techniques, collaborative filtering (CF) [1], usually leverages previous user-item history to generate lists of recommendations. However, for new items which have not been visited or rated by any users, we do not have sufficient information to apply CF strategy. Such recommendation problem is called *cold start* problem: given a set of users and new items, recommend a list of new items to the users they may be interested in.

One typical way to tackle the cold start problem is *content-based recommenda-*

tions. The user will be recommended items similar to the ones the user preferred in the past. However, there are situations that we need to recommend new items to new users. Suppose a user visits Amazon for the first time, she is browsing one product, and the web site tries to suggest more products she may be interested in. There is no personal history information about this user as she is new to the web site, therefore it is difficult to apply collaborative filtering directly. The web site may suggest other similar items because she may consider other alternative products. The site may also suggest items that other users purchase together with the product she is browsing, because such items may be complementary to the product. They may also combine two strategies to make recommendations. More difficultly, there may be new products that the web site wants to promote. In such a situation, we cannot directly exploit co-purchase history to recommend complementarily, but only look at the product similarity. Mamy aspects of products can be used for similarity measure, e.g., products that are produced by the same factory may also be interesting to users, or the products that have similar function can be of interest.

How can we decide which aspect is more important in this new product recommendation than others? In this chapter, we propose to learn the importance of different aspects from the implicit user co-purchase history . By way of illustration, if we find users purchase products from the same specific manufacturer frequently, we can suggest new products from the same manufacturer too; if we find people purchase products together for some function, we may recommend products for similar functions. We think this way may work better than naively comparing two products.

In this chapter, we study such a cold start problem exploiting implicit user generated data. More specifically, we focus on cold-start recommendations for lecture videos using co-view information based on previously seen lecture pairs. Formally, we define our problem as follows:

Problem 1 (Cold Start Lecture Recommendation Problem:) *We are given a set of training lectures Q, and a co-viewed pairs set $P = \{(l_1, l_2, f) \mid l_1, l_2 \in Q\}$, where f is the frequency that l_1, l_2 were co-viewed together. The test set, T contains lectures without any viewing history and we are required to recommend lectures, $R_q \subset T$ for each query lecture $q \in Q'$, $Q' \subset Q$.*

This problem simulates the scenario in which recommendations are to be made for a new user or a new lecture where no co-viewed history information is available.

We adopt a content-based approach for the cold-start scenario where co-view information is used to learn the feature weights for ranking lectures for the recommendation task. We use the co-viewed lecture pairs to form training instances for a supervised learning setup. Support Vector Machines were used where the learnt feature weights indicate the importance of each lecture attribute for recommending lectures in the cold-start scenario. Our solutions based on the above strategies performed on par with the top-performing baselines.

The remainder of this chapter is organized as follows: Section 4.2 briefly introduces related work in recommender system research field. Section 4.3 describes the data attributes in our recommendation scenario. Section 4.4 present the features we design and extract. Section 4.5 describes our solution design using SVM classification framework. Section 4.6 shows our experimental results. Section 4.7 concludes the chapter.

4.2 Related Work

Various techniques for recommender systems have been proposed in the last few years. In this section, we do not intend to cover every aspect and detail of related work in the research area of recommender systems, but briefly introduce common background and focus on literature review of research problems most closely related to our work. We refer interested readers for further understanding

to the introduction in the *Encyclopedia of Machine Learning* [62] and the excellent survey [1].

Recommendation strategies can be broadly classified into collaborative filtering and content-based strategies. We briefly describe the basic ideas behind these approaches here.

Collaborative filtering (CF) methods use previous item-user history to generate lists of recommendations. For example, in movie recommendations, CF strategies use movie ratings previously submitted by other users to predict the rating a user might assign to a movie based on user-similarity or movie-similarity [1]. CF can be further divided into *neighborhood-based* and *model-based* approaches [62]. In neighborhood-based CF, suppose a system tries to recommend a list of items to user u, it would first find top-k most "similar" users to u, and estimate ratings of items based on a weighted combination of those k neighbors' ratings. Systems usually computes similarity between users from their item rating history. On the other hand, model-based CF generates recommendations by estimating parameters of statistical models for user ratings. Matrix factorization (MF) is currently the state-of-the-art method in this class of techniques [44]. MF assumes that similarity between users and items is simultaneously induced by hidden lower-dimensional vector of factors inferred from item rating patterns [43]. MF is popular because of its good scalability and predictive accuracy in Netflix prize [1]. However, there is no theoretical proof that this model can be generally applied in other recommendation scenarios.

Content-based methods are common in addressing the `cold start` problem where ratings and preference information is unavailable or sparse. Many content-based methods focus on recommending items with textual content. As such, techniques in information retrieval (IR) can be applied conveniently [4]. Many other approaches treat such a problem as a classification task, using techniques like

[1] http://www.netflixprize.com/

naive Bayes classifier, k-nearest neighbor (kNN), decision tree, etc [71].

Recently, hybrid strategies are being used to leverage the benefits of both collaborative filtering and content-based strategies. For example, to tackle the *cold start* problem, Gantner, et al. [29] used collaborative information to compute similarity between existing items or users using matrix factorization, and then proposed mapping techniques like a linear combination of various attributes of new items to fit content into same model.

To tackle *cold-start* problem, [82] and [25] uses feature weighting strategy in recommender systems. [82] proposed a feature weighting method to combine various movie attributes like release year, language, director, etc. This method uses normalized number of users who are interested in two movies as a human judged similarity between a movie pair. Then they solve regression equations constructed between features and paired similarity to obtain weight values. [25] employed a similar approach of combining multiple features in a weighted linear model for citation recommendation of academic papers. In order to find weights of different features between papers, e.g., text similarity and co-citation, the authors applied a coordinate ascent method. As opposed to the regression framework adopted by them, we formulate the attribute-weight learning problem in a classification framework for cold-start recommendations.

4.3 Description of Lecture Attributes

We target recommendations for lecture videos from `videolectures.net`. `Videolectures.net` is an open-access repository of educational lectures [2]. Lectures given by prominent researchers and scholars at conferences and other academic events are made available on this website for educational purposes.

Figure 4.1 denotes a snapshot of the system at `videolectures.net`. It was

[2]http://videolectures.net/site/about/

captured at August 2010 when our data set was collected. We indicate in this figure some of the information available with lectures on this website. Most lectures on this website contain information on the language in which the lecture was given, content of the slides, the category (discipline-area) of the lecture, etc. Sometimes, additional information such as the description of the event (such as conference, workshop) in which the lecture was given and author affiliation is also available. Along with the lecture, authors and event attribute information, the data also includes user feedback as pairs of lectures that were frequently co-viewed in the past.

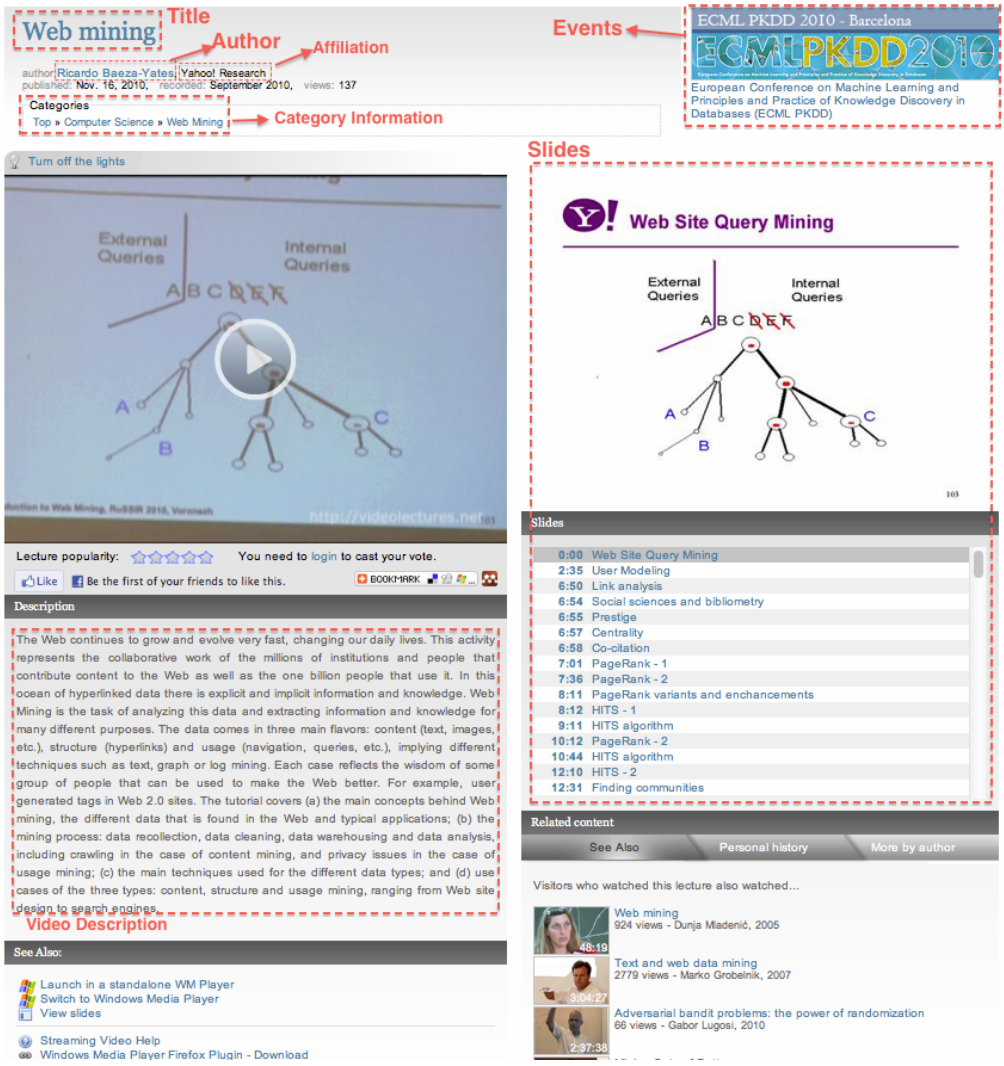
The data set is collected and made available by [2]. In the following we briefly summarize attributes information available with the data.

1. **Author Attributes:** We have authors' full name, e-mail address, each author's homepage, gender of the author, and his/her organization. The organization usually means affiliation where the author works.

2. **Lecture Attributes:** For each lecture, it contains the lecture's title, and published date. It also contains a description attribute, which is usually the abstract of the lecture if it is paper presentation published in some conference venue. Slide titles are also included in this data if available. Moreover, lecture is manually grouped into a predefined set of event types, e.g., keynote, tutorial, invited talk, thesis proposal, etc. Event information is also included here, e.g., a paper is published in *SIGKDD 2013 Conference*. Besides, there is another manually predefined taxonomy attribute called *category*. More details will be introduced in next item. Language information is also included here, by default it is English. Finally, each lecture has a number of views since the day it was published online to the day the data snapshot was taken.

3. **Categories:** Category attribute is used to represent the lecture's subject or research field information. The web site `videolectures.net` uses a scientific

Figure 4.1: Lecture video attributes



taxonomy same as Wikipedia to represent lectures' categories. The taxonomy is represented as a direct acyclic graph where one category can have multiple parent categories. Each category must have a parent category except the root category. Each lecture can be assigned to several categories. For example, a lecture may be assigned to "Computer Science" and more detailed "Data Mining" categories.

4. **Pairs:** We have records of pairs of lectures viewed together. The pairs of lectures are not necessarily viewed consecutively. They were detected with as

least two distinct cookie-identified browsers. The frequency of each pair was viewed together is also included.

4.4 Feature Description

The intuition behind this cold-start problem is that we want to recommend lectures that are similar to the one being viewed. The pairs information available for this recommendation problem indicates the frequency with which a given lecture pair was co-viewed. This information is very significant in understanding the features that a pair of lectures that tend to be co-viewed often share. More specifically, our assumption is that the more frequently two lectures were co-viewed, the more similar they are. For instance, it is reasonable to expect that a highly co-viewed pair of lectures are in the same language and perhaps in the same category. Similarly, a pair that is co-viewed frequently is likely to be on related topics such as two lectures presented in the same conference or two parts of a tutorial on a topic. It is also intuitive to expect the co-view frequencies of lectures belonging to diverse categories such as Graph Theory and Ecology to be small. Based on the above intuitions, we designed the set of following features to measure the similarity between two lectures in terms of their attributes.

1. **Co-author similarity** This feature indicates whether two lectures have the same author. It has a value 1 when two lectures share the same author and 0 otherwise.

2. **Type similarity** This feature has a value 1 when two lectures share the same type and 0 otherwise. Example lecture types include lecture, keynote, thesis proposal, tutorial etc.

3. **Language similarity** has a value 1 when the two lectures are in the same language and 0 otherwise.

4. **Event similarity** A value of 0 or 1 indicates whether the two lectures belong to the same event such as conference, workshop series etc. In addition to using the above boolean-valued feature, we used the description fields associated with events to compute a similarity value using the cosine similarity function. This score is meant to capture events that are similar though not the same. For instance, the conferences ECML and ICML are related despite being distinct venues since they are both machine learning conferences. Similarly, lectures belonging to the same conference venue but presented in different years are related.

5. **Category similarity** The category information pertains to the subject area assigned to a lecture. Connections between categories are captured via a directed graph can be used to compute similarity. For instance, if two lectures are assigned the categories “Computer Science" and “Graph Theory", they share some commonality since “Graph Theory" is a sub-category of “Computer Science". To capture this aspect, we used four different binary indicators for capturing category similarity between two lectures l_1, l_2:

   - *C1:* 1 if $l_1.categories \cap l_2.categories \neq \emptyset$ and 0 otherwise.
   - *C2:* 1 if $l_1.categories \cap l_2.parent\ categories \neq \emptyset$ and 0 otherwise.
   - *C3:* 1 if $l_1.parent\ categories \cap l_2.categories \neq \emptyset$ and 0 otherwise.
   - *C4:* 1 if $l_1.parent\ categories \cap l_2.parent\ categories \neq \emptyset$ and 0 otherwise.

6. **Text similarity** The name, titles and description fields of a given lecture have textual content. We represent these fields using TFIDF [58] vectors and use the cosine similarity of the corresponding fields of two lectures to compute these features.

7. **Topic similarity** We use LDA [9], a popular tool used for modeling documents as topic mixtures. The generative process in LDA expresses each

document in terms of its topic proportions. We model the training set of lectures (name+description+titles) using 1000 topics and obtained the topic proportions for each lecture. Similarity between a pair of lectures can be computed using the cosine similarity between the topic vectors or by measuring the overlap among the top topics from each lecture. We used Jaccard Coefficient [58] to compute the similarity score based on the overlap among the top-10 topics of the two lectures.

8. **Affiliation similarity** The author affiliation information is also available with lectures. We compute the affiliation similarity between two affiliations with the Jaccard similarity measure on the set of words describing the affiliation.

4.5 Learning attribute weights for pairwise prediction

Support Vector Machines (SVM) is a discriminative supervised learning approach widely used for classification and regression problems in several areas. For binary classification where the set of class labels is restricted to +1 and -1, the SVM learns a maximally separating hyperplane between the examples belonging to the two classes based on the training data. During testing, the distance between a given instance and this hyperplane is computed and used to assign a prediction label.

We formulate the recommendation task for the cold start scenario as a binary classification problem. We treat the co-viewed lecture pairs available in the training data as positive examples for the classification problem. Negative instances for training the classifier are obtained by randomly selecting lecture pairs that were never co-viewed (in the training data). The features described in Section 4.4 were used to train a SVM classifier. The data includes query lecture (from say, the set Q)

for which recommendations are to be predicted from the set of given test lectures (say, set T). We used each $q \in Q$ to form a pair with each $t \in T$ and score the pair using the trained SVM classifier, namely the distance from this lecture pair instance to the hyperplane. The final list of predictions for each query is obtained by sorting the pairs based on these scores and choosing the test lectures corresponding to the top pairs.

When trained with the linear kernel option, SVMs learn a set of weights that satisfy the maximum number of constraints of the following form imposed by the training data:

$$y_i(\vec{w}.\vec{x}_i - b) \geq 1 - \epsilon_i, 1 \leq i \leq n$$

In the above formula, i is the index over the training examples, $\vec{x}_i$ pertains to the features of a given example, y_i its label (+1 or -1) [15]. In our case, the feature values refer to similarity values based on different attributes of a given lecture pair. That is, as part of learning the classifier, we are in effect, learning a scoring function for lecture pairs (l_i, l_j) based on a linear combination of individual attribute similarity values such that

$$score(l_i, l_j) = \sum_{f=1}^{F} w_f \times sim_f(l_{fi}, l_{fj})$$

where w_f indicates the weight assigned to the similarity value based on a particular attribute f of the given lecture pair.

4.6 Experiments

4.6.1 Data Description

For the experiment, we collected a data set published by ECML/PKDD 2011 conference [2]. The data set was captured in August 2010 from the video lecture

web site [3]. The data set consists of meta information about lectures and authors. It also contains view statistics about lectures. There are a total of 8105 video lectures, 8094 authors, 366 categories, and 520 events in this data set.

Lectures are split into training and test subsets by publication date. Those lectures published before July 01, 2009 are put into *old* lectures as training data set, and lectures published after July 01, 2009 into *new* lectures as test set. This leads to a split of about 85% for the training and 15% for the test data set. We will try to recommend *new* lectures for a set of query lectures slected from training set. Accordingly, co-viewed pairs from *old* lecutres are also provided in the training data set. Statistics of this data set is shown in Table 4.1.

Table 4.1: Statistics of Data Set

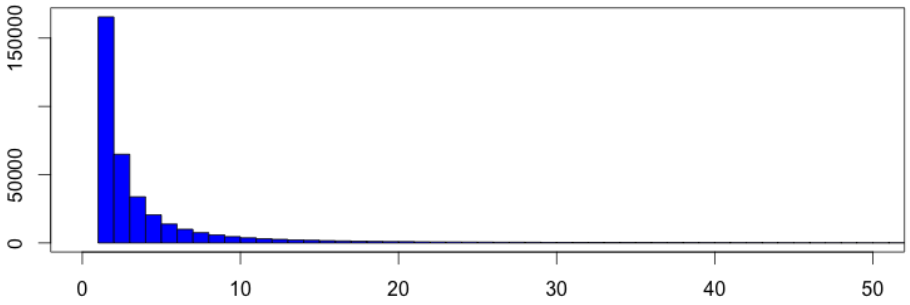
Data Statistics	**Number**
Total number of lectures in the data set	8,105
Number of lectures in training data set	6,983
Number of lectures in test data set	1,122
Number of co-view pairs in training set	363,880
Number of categories	366
Number of events	520

Figure 4.2 illustrates histogram of co-view pair frequency in training data set. While Figure 4.2a shows histogram of values truncated to 50, Figure 4.2b illustrates histogram of pair frequency larger than 50 which is truncated to 500. These figures demonstrate that the distribution has obvious long tail shapes.

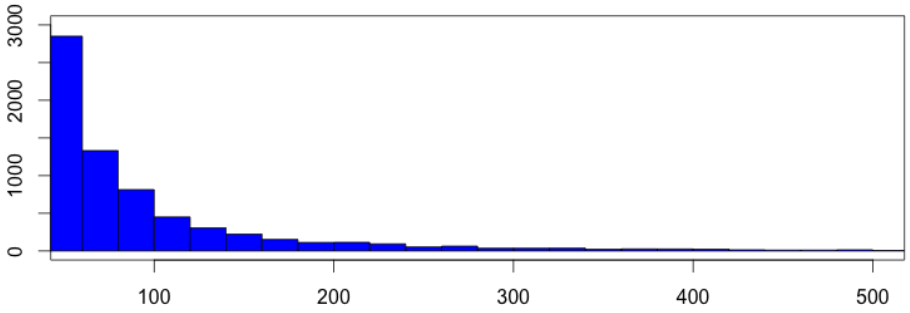
4.6.2 Evaluation Metric

Recommendation results are usually measured through user satisfaction surveys and analysis. However, for our problem we need a quantitative measure that can evaluate solutions meaningfully. We exploit the metric of mean average R-precision score (MAR_P) for evaluating recommendation performance. MAR_p is a mean

[3] http://videolectures.net



(a) Co-view pairs frequency truncated to the level of 50



(b) [Co-view pairs frequency larger than 50, and truncated to the level 500

Figure 4.2: Histogram of co-view pairs frequency

value over all queries R defined as:

$$MAR_p = \frac{1}{|R|} \sum_{r \in R} AvgRp(r)$$

Here average R-precision for a single recommended ranked list is given by

$$AvgRp = \sum_{z \in Z} \frac{Rp@z(r)}{|Z|}$$

where

$$Rp@z(r) = \frac{|relevant \cap retrieved|_z}{|relevant|_z}$$

the R-precision has some cut-off length z where $z \in \{5, 10, 15, 20, 25, 30\}$.

The R-precision metric is more apt to our *cold start* problem scenario, because it adjusts to the size of the set of relevant documents. By averaging over a set of different $Rp@z$, we can take into account ranking of relevant items and improve the ability of differentiating between similar solutions. MAR_p metric is better than MAP (mean average precision) measure, because MAP does not consider absolute ranking positions of recommended items. Different permutations of relevant items in the recommended list do not affect MAP score. Moreover, if ranking order does not need to be strict for top-n item recommendations [87], the granularity of ranking can also be adjusted. Here the granualiry of 5 was chosen according to the ranking-recall influence on recommender system evaluation [2].

4.6.3 Experiment Results

4.6.3.1 Recommendation without Co-view Information

Cold-start problem is usually tackled with *content-based* recommendation methods. In content-based methods, items are recommended that are "similar" to users or to items that users have previously preferred. In this lecture recommendation scenario, our goal is also to recommend lectures "similar" to the given query lecture. As we design a framework to improve recommendations by utilizing implicit user generated data like co-view information, in this subsection we demonstrate the results of baseline methods that only exploit lecture content without co-view information.

The content-based recommendation method is closely related to information retrieval (IR) [1,57]. We simply convert this lecture recommendation setting into an IR problem and exploit IR techniques for recommendation purpose. More formally, given a query lecture q from query set Q', we try to retrieve a set of lectures R_q

that are most relevant to q from the test lecture set T.

We concatenate lecture title, description, author names, author affiliation, event information, category information, and available slide titles as a document for each lecture. Then we explore several text retrieval models over these documents. Two basic text retrieval models, vector space model and language model [57], and a few variations are studied in our experiments. These models are briefly summarized in Table 4.2. We elucidate more details in the following:

Vector Space Model

Vector space model is an algebraic model for representing a set of documents as vectors in a common vector space [57]. It is widely used in information retrieval operations ranging from scoring documents on a query, document classification and document clustering. To compute similarity between two documents d and q, we can leverage the standard *cosine similarity* of their vector representations $\vec{v}_d$ and $\vec{v}_q$ as follows:

$$sim(q, d) = \frac{\vec{v}_d \cdot \vec{v}_q}{|\vec{v}_q||\vec{v}_q|}.$$

We study three methods in vector space model in our problem setting: term frequency-inverse document frequency (tf-idf), topic model from LDA [9], and a combination of the two methods.

- **tf-idf:** In tf-idf, each document d is represented as a vector with one component corresponding to each term in the predefined dictionary. The weight of each component in the vector, or for the term t in document d, its weight is assigned as:

  $$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t,$$

  where $\text{tf}_{t,d}$ is the number of occurrences of term t in document d, and $\text{idf}_t = log\frac{N}{\text{df}_t}$, N is the total number of documents and df_t is the number of documents that contain term t.

- **LDA:** As described in Section 4.4, in LDA each document can be modeled as topic mixtures, which are represented as vectors of topic distribution. We directly leverage this topic vector as the document representation, and compute cosine similarity based on the topic vectors.

- **tf-idf + LDA:** As each document can be potentially represented in two vectors, for each pair of document d and q, we can also have two similarities. By introducing a small smoothing factor $\lambda < 1$, we combine the two similarities together:

$$sim(q, d) = \lambda sim_{\text{tf-idf}}(q, d) + (1 - \lambda) sim_{\text{LDA}}(q, d).$$

Language Model

In information retrieval, the language modeling approach models the idea that a query q is generated by a probabilistic model based on a document d [57, 97], denoted as $P(q|d)$. In order to rank documents, we are interested in estimating the posterior probability $P(d|q)$, which is derived based on Bayes' formula from

$$P(d|q) \propto P(q|d)P(d)$$

where $P(d)$ is the prior probability for document d and is often treated as uniform for all documents. Therefore ranking documents for a given query is also a problem of ranking the probability of $P(q|d)$. We consider unigram language model in which the probability of each word appearing in a document only depends on itself. Suppose a query q is composed of a set of words $\{w_1, w_2, \cdots, w_m\}$, then we have

$$P(q|d) = \prod_{i=1}^{m} P(w_i|d),$$

where $P(w_i|d)$ is the probability of word w_i in language model of document d.

One way to approximate word probabilities is using maximum likelihood estimation (MLE) as $P(w_i|d) = \text{tf}_{w_i,d}/L_d$, where $\text{tf}_{w_i,d}$ is the frequency of word w_i in document d, and L_d is the number of tokens in document d. However, there may be situations when some words have never appeared in some documents but may be needed in a query. In such cases, if we estimate $P(w|d) = 0$, we will get a probability of zero for the query too. Researchers have studied methods to ease such problems and introduced many smoothing techniques. We will explore a few of them for our experiments in this subsection.

- **MLE:** One common way of smoothing language model is to leverage word probability from the perspective of the whole collection of documents, we simply call such a method as MLE [97]. The probability is calculated as:

$$P(w|d) = \lambda P_{ML}(w|d) + (1 - \lambda)P_{ML}(w|Coll),$$

  where $P_{ML}(w|d)$ is the maximum likelihood estimate of word w in document d, $P(w_{ML}|Coll)$ is the maximum likelihood estimation of word w in the document collection, and λ is a smoothing constant that has value less than 1.

- **ClusterMLE:** Researchers have also studied another smoothing technique based on document clusters [55, 61]. Instead of using word probability from the whole document collection, researchers first carry out clustering over documents, and introduce the word probability based on such clusters into smoothing. More specifically, the word probability is calculated as:

$$P(w|d) = \lambda P_{ML}(w|d) + (1 - \lambda)[\beta P_{ML}(w|Cluster) + (1 - \beta)P_{ML}(w|Coll),$$

  where $P(w|Cluster)$ is the maximum likelihood estimate of word w in the cluster that document d belongs to, and β is another constant factor with value

Table 4.2: Model specification in experiments

	Model
tf-idf	$sim(q,d) = \vec{v}_d \cdot \vec{v}_q$, using TF-IDF vector of document
LDA	$sim(q,d) = \vec{v}_d \cdot \vec{v}_q$, using topic distribution vector of document
tf-idf + LDA	$sim(q,d) = \lambda sim_{TF-IDF}(q,d) + (1-\lambda) sim_{LDA}(q,d)$
MLE	$P(w\|D) = \lambda P_{ML}(w\|D) + (1-\lambda) P_{ML}(w\|Coll)$
ClusterMLE	$P(w\|D) = \lambda P_{ML}(w\|D) + (1-\lambda) P(w\|Cluster)$ $= \lambda P_{ML}(w\|D)$ $+ (1-\lambda)[\beta P_{ML}(w\|Cluster) + (1-\beta) P_{ML}(w\|Coll)]$
MLE + LDA	$P(w\|D) = \lambda(\beta P_{ML}(w\|D) + (1-\beta) P(w\|coll))$ $+ (1-\lambda) P_{lda}(w\|D)$

less than 1 that we can adjust the weight of cluster based word probability.

- **MLE + LDA:** We also introduce LDA into language model and explore the topic effects in smoothing. We calculate the probability of word w appearing in document d under topic modeling, represented as $P_{LDA}(w|d)$, and combine it with language model as:

$$P(w|d) = \lambda[\beta P_{ML}(w|d) + (1-\beta) P_{ML}(w|Coll)] + (1-\lambda) P_{LDA}(w|d).$$

Text Retrieval Results

We filter out documents that have less than eight words, leading to 5,236 documents, and 363,880 co-view paris. We randomly divide the documents into five folds, using one fold as a test set, and remove all pairs between test and training set. Evaluation is carried out using MAR_p for the cut-off length $z \in \{5, 10, 15, 20\}$. For LDA and document clustering, we use Mallet tool kit [60]. We use 160 topics in LDA. After tuning, We set 0.66 for λ in tf-idf + LDA, set $\lambda = 0.11$ for MLE model, set $\lambda = 0.1, \beta = 0.01$, and set $\lambda = 0.21, \beta = 0.11$ for MLE+LDA model. The final results after cross validation are shown in Figure 4.3.

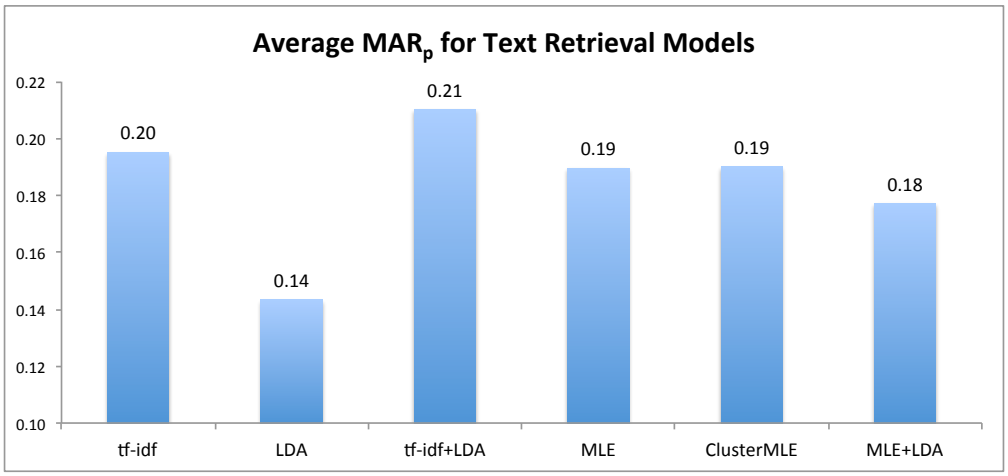


Figure 4.3: Results o fvarious models

We can see that tf-idf simply works almost the best. Leveraging LDA topic vector directly is almost the worst. However, combining tf-idf and LDA can slightly improve the results. MLE model is a little worse than tf-idf in our scenario. Combining LDA with MLE does not work as well as tf-idf either. These results demonstrate that tf-idf is still the most efficient method, and topic modeling can be leveraged to slightly improve the recommendation results.

4.6.3.2 Recommendaiton Incorporating Co-view Information

We incorporate co-view information and formalize the recommendation problem into a classification framework using SVM classifier. For training the SVM classifier, from the training set P we filter out pairs that occurr with a frequency less than 5% (for either lecture in the pairs):

$$P' = \left\{ (l_1, l_2, f) \mid (l_1, l_2, f) \in P \,,\ \frac{f}{S(l_1)} \geq 0.05 \ \vee \ \frac{f}{S(l_2)} \geq 0.05 \right\},$$

where

$$S(l_1) = \sum_{(l_1, l_i, f_{1i}) \in P} f_{1i}, (l_2) = \sum_{(l_j, l_2, f_{2j}) \in P} f_{2j}.$$

These are assigned the class label +1 as positive instances. For negative instances, we randomly select lecture pairs of a comparable size to P' that do not appear in the training pairs set P. In total we have a balanced data set with about $40,000$ pairs for training the classifier. We use the SVMLight [38] implementation provided by Joachims. We set the margin-loss penalty parameter C to 10 after experimenting with values between 0.1-100. The performance on a validation set is the best for C values ranging between 5-20. To show the stability of feature weights we show their mean and variance over five-folds of training runs in Table 4.3.

Table 4.3: Feature Weights Learnt By SVM

Feature	Mean	Variance
Co-author similarity	1.895	0.050
Type similarity	-0.087	0.005
Language similarity	0.015	0.003
Event similarity (exact match)	0.779	0.110
Event description similarity	1.421	0.118
Category similarity $C1$	1.889	0.038
Category similarity $C2$	0.114	0.015
Category similarity $C3$	-0.058	0.019
Category similarity $C4$	0.268	0.195
Name TFIDF similarity	8.555	0.125
Description TFIDF similarity	-1.412	0.094
Slide Content TFIDF similarity	-0.360	0.100
All Text fields TFIDF	9.729	0.273
Jaccard similarity based on LDA Top 10 topics	0.329	0.016
Affiliation similarity	0.705	0.189

As shown in Table 4.3, the positive weights for some features such as co-author similarity, event similarity and LDA topic overlap support our intuitions on what attributes are common in lectures that are co-viewed frequently. The negative weights for description and slide content similarity is surprising. We reason that this is possibly due to the fact that a large number of lectures in the training data have empty values for these fields. Similarity based on the concatenated field combining the name, description and slide content fields and the name similarity

Table 4.4: Performance with different SVM settings

Model	MAR_p
Classification	0.2517
Classification without feature selection	0.2403
Regression	0.1100
Ranking	0.1697

fields have high positive weight values that are not surprising. Videos belonging to the same event such as lectures from a course series are likely to share a lot of content similarity in their name fields and are also likely to be viewed together. We discard features with negative weights and re-train the classifier based on the remaining features. This leads to a slightly better final result shown in Table 4.4.

The classification setup treated all paired lectures uniformly as positive instances. However, since it is likely that lectures with higher co-view frequencies are most similar, we also try unequal weighing strategies based on co-view frequencies as a ranking or regression problem using SVM. With similar features in classification, rather than +1 or -1 as class label, we define different target values based on co-view frequencies of pairs for regression and ranking setup. For regression setting [38], the target similarity value of a pair instance (l_1, l_2, f) is defined as $s = \frac{f}{S(l_1)+S(l_2)}$ and normalized later. In ranking setting [39], for each query lecture video q, the target value is defined as pairwise preference according to co-viewed frequency, namely, in training set for each video p paired with q, the larger co-viewed frequency (p, q) has, the higher ranking it stands. Table 4.4 shows that our preliminary experiments where a regression and ranking formulation was adopted performed worse than classification, but further experiments on understanding this aspect are required.

4.7 Conclusions

In this chapter, we study a cold-start problem in the context of video lecture recommendation. We study closely the relative features, including various meta data

like the author information, the categories videos belong to, and most importantly the co-viewed feedback from users. Using such user feedback, we formalize this recommendation problem into a classification framework. SVM is leveraged to learn weights of all features in determining the new videos' similarities to query videos. We obtain a MAR_p score of 0.25456 in this problem, which is significantly better than baseline method like TF-IDF.

We need further study to understand the performance difference between SVM classification and regression or ranking formulation. Further, in our experiments, we fit a single model over all lectures in the training data. It is possible that the lectures can be somehow clustered so that a different model is learnt for each cluster.

Chapter 5

Exploiting Social Connections for Team Recommendation

In the previous chapter, we describe our work to demonstrate how implicit user feedback can benefit cold-start recommendation tasks. In this chapter, we will study a different recommendation scenario, the team recommendation problem, and present our solution framework with benefits from social relations among users. Specifically, we investigate the team recommendation problem in IT strategic outsourcing services, and study how we can leverage social connections to combine both individual strength and team features together for recommending the optimal team.

5.1 Introduction

A competent team is critical for the success of any activity that requires teamwork. In sports, e.g., basketball, football, and volleyball games, players cannot win if they do not collaborate well as a team. In military, missions can not be accomplished without highly specialized and coordinated teams. Similarly, teams are essential to the business success of companies in industry, no matter whether they are startups

or large international organizations. Team recommendation or formation using a computational approach has recently drawn a lot of attention from operation research [7, 17, 98] and data mining communities [23, 47]. In this chapter, we study the team recommendation problem in the real business world, specifically, IT strategic outsourcing services. To the best of our knowledge, this work is the first of its kind in a realistic business scenario.

IT strategic outsourcing (SO) is the process of contracting IT related business functions, for example, help desk services, cloud services, and network management, to external service providers. Companies seeking to outsource typically issue a request for proposal (RFP) to service suppliers. Interested providers can respond to the RFP by submitting a detailed solution, typically in a very competitive bidding process. The solution needs to address the potential client's requirements by detailing the bidder's technical capabilities, price of services, estimated completion time, etc. Design of IT service solutions, especially for large strategic outsourcing deals with billions of dollars, is a very complex process. It requires a wealth of knowledge and expertise not only in external aspects such as client's business, competitor's strengths, and insights into successful win strategies, but also in internal company aspects such as capabilities of available technical service offerings, costing and pricing guidelines, etc. All of these factors need to go into designing a winnable solution that can achieve the desired profitability.

The time window for submitting service solutions to clients can be very short. Especially, in highly competitive markets, service providers have to make quick responses to client's RFP. Locating experts with right skills and forming a competent team to act swiftly thus become very critical. In this work, we refer to such a team of experts as a *deal team*, and the task of finishing a service solution as a *project*. Experts take different job roles in a deal team, such as project leader, solution architect, sales manager, development manager. Solution architects also have more detailed job roles according to their specialized service components.

Experts' individual expertise and experience play an important role in accomplishing the project. It is also known that how effectively they communicate and collaborate with one another can affect team performance. For example, Huckman et al. [35] show that in an Indian software services firm, the level of team familiarity (i.e., the average number of times that team members have worked with one another) has a significant positive effect on team performance (i.e., the number of defects in software). For IT strategic outsourcing services, in large international organizations, experts may come from different business units or are physically located in different offices. Understanding their work collaboration history and connection strength is important to build a competent team.

Social connections can be quite beneficial in measuring and evaluating team strength. Researchers propose the concept of *communication cost* to represent the strength of teams [41, 47]. The smaller the cost, the better a team. They first construct a weighted or unweighted social graph based on connections among potential candidates, define the *communication cost* of potential teams based on network characteristics. Various network characteristics can be leveraged for the cost definition, e.g., the diameter of a subgraph composed of team members.

In this chapter, we propose a general team recommendation framework that considers both individual strength and team features. We extract individual strength from people's work experience, and derive potential teams' characteristics from social networks built on various social activities, including their collaboration history and other online interactive activities. Different from all the existing work, we leverage the outcomes of historical projects (win or not win) to identify important features in team recommendation using a machine learning approach. A team quality metric is proposed by a linear weighted combination of these features. Our contributions are summarized in the following:

- We propose a team recommendation framework which considers both individual and team level characteristics for team recommendation. Different from

existing work which only use a single type of team feature, our framework can easily incorporate multiple types of team information. The proposed recommendation framework is general and can be applied to different domains.

- When historical projects along with their outcomes exist, a machine learning approach is applied to learn the weight of each feature. A team strength score is proposed using the weighted features.
- A heuristic search algorithm is applied to find the approximate optimal team.
- We apply our framework in a real business setting, i.e., IT strategic outsourcing services. The results show that our framework works well in practice.
- To evaluate our recommendation algorithm, we apply it to public DBLP data set for academic team recommendation. The results demonstrate the effectiveness of our algorithm compared with existing approaches.

5.2 Related Work

A lot of research efforts have been recently devoted to the problem of *team recommendation* or *formation*[1], for example, forming teams in academic paper authoring [42, 47], movie acting [42], and engineering [17, 35].

In data mining community, Lappas et al. [47] were the first to study the problem of finding a team of experts from a social network. A communication cost is defined to measure how effectively team members can collaborate: the lower the cost, the better the collaboration. The authors studied two types of communication cost for team formation: the diameter cost, which is the largest shortest path between any pair of nodes in the subgraph formed by a team, and the minimum spanning tree among the team members. Two heuristic approximation algorithms were designed

[1]Since both team recommendation and team formation are about finding a team that can cover a set of specified skills or job roles, we use these two terms interchangeably

specifically. Kargar et al. [41] studied two other different communication costs: the sum of distances, which calculates the sum of shortest distance between pair of experts, and the cost involving a leader role in the team. Kargar et al. [42] later proposed a bi-objective team formation framework which considers both personnel cost and communication cost among team members.

However, the above work only considers one single type of communication cost (or collaboration) among a team. There are other social connections in work practice. Team members in the same division may communicate better or more easily than those from different divisions. Although people may have not worked together on a project before, they can already be friends and know each other well. Such factors can also be employed for team formation. Besides, existing work cannot handle multiple team costs from different sources. This restricts their approaches from expanding to other applications that require different cost functions. In this work, we instead propose a general framework that can incorporate arbitrary types of team costs or features.

Datta et al. [23] studied similar team recommendation problem by leveraging academic knowledge networks. The authors consider team cohesion as an important factor in ranking teams. Various types of social graph and clustering coefficients can be used to calculate team cohesion. Researchers also studied several social factors in team recommendation [13, 22, 88] They represent socio-semantic interactions in three graphs: the competence network from a bipartite graph of users and concepts, the social connections from interactions among users, and the team network hyper-graph from joint appearance of users and concepts. However, the empirical evaluation from such work is only carried out with regard to time factor, e.g., how long it will take to compose a needed team. We cannot see any evaluation metric about how teams can finish some specific tasks.

Our problem is also related to expert search. SmallBlue [50] demonstrates one such study. SmallBlue is a social networking application in business intelligence. It

leverages both public user profile information and private messages to analyze social network within a company. The purpose of this system is to answer the questions of "who knows what" and "who knows whom". Its functions include mining expertise, retrieval about experts, and visualization (via ExpertiseNet), it also provides social network analysis for people's collaboration and expert recommendation. However, such a system does not have the function of team recommendation.

5.3 Team Recommendation

In this section, we first give the formal team recommendation definition and then introduce a general team recommendation framework by combining individual strength and team features. Two types of team features are considered. We apply a machine learning approach to learn the weights of different features and propose a team quality metric. We finally introduce Max-logit, an equilibrium selective learning algorithm, to find teams in a combinatorial solution space.

5.3.1 Problem Definition

Let $P = \{p_1, p_2, \cdots, p_n\}$ be a set of people and $S = \{s_1, s_2, \cdots, s_m\}$ denote a set of roles. Each person can work as a set of different roles, denoted as $H(p_i) \subset S$. If $s_j \in H(p_i)$, we say that person p_i can work as role s_j. For each role s_j, we denote the set of people who can work in s_j as $C(s_j) \subset P$. Given a project that requires a set of roles $S' \subset S$, we need to find a team $T \subset P$ so that every role $s_j \in S'$ will be covered by a person from T, that is, for each $s_j \in S', C(s_j) \cap T \neq \phi$.

The goal of team recommendation is to find the *best* team that can cover the required role set. However, how to measure team competence varies a lot in different applications. Multiple types of costs or features may co-exist and affect the team performance together. For example, a person's expertise and experience may play an important role in determining her chance of joining a team. interaction and

collaboration efficiency among team members need also be taken into consideration. We therefore propose a general framework that can combine multiple types of features in team recommendation.

Without loss of generality, we define the following three functions.

Definition 1 (Individual Feature Function) $F : P \rightarrow \Re^K$. *Each person has K predefined individual features, such as experience and expertise.*

Definition 2 (Team Feature Function) $G : \boldsymbol{P}(P) \rightarrow \Re^L$. *A team composed of a subset of P has L team features, such as team closeness and social connection strength.*

Definition 3 (Team Strength Function) *Given a team T of individuals for a project S', its team strength defined as:*

$$TeamStrength(T) = \frac{1}{|T|} \sum_{p_i \in T} \overrightarrow{W_1} F(p_i) + \overrightarrow{W_2} G(T),$$

where $|T|$ is the cardinality of team T or the size of the team, $\overrightarrow{W_1}$ is the weight vector for individual features, and $\overrightarrow{W_2}$ is the weight vector for team features. All the weights are constrained to be non-negative. That is, we assume these features either have no effect or positive effect on team strength.

We define our team recommendation problem as follows.

Problem 2 (Team Recommendation Problem) *Given a project S', the individual features F for all people, team features G for all possible team formation, and weights $\overrightarrow{W_1}, \overrightarrow{W_2}$ for all features, find a team T so that each role in S' will be covered and only covered by one person, each person will cover at least one role, and $TeamStrength(T)$ is maximized.*

Note that we define team strength function instead of cost function. While previous work tries to minimize team cost, our objective is to maximize the team strength.

5.3.2 Feature Description

The individual and team features used in this chapter are summarized in Table 5.1. While most individual features are extracted from project history data, the team features are from collaborations graph in project history data and various other social connections.

Table 5.1: Feature List

Feature	**Description**	**Source**
Individual experience	Number of participated projects in history	Project History
Individual win experience	Number of winning projects in history	Project History
Individual win rate	Winning rate in history	Project History
Individual role experience	Number of participated projects as specific role in history	Project History
Team closeness	Closeness from work collaboration graph	Project History
Social connection	Various connections in the team	Social graph constructed from work practice

5.3.2.1 Individual Features From Project History

One approach to measure individual capability is to look at a person's project history. Intuitively, the more projects a person has worked on and achieved successful results, the more capable she could be. Therefore we extract a few individual features from people's project experience:

- *Experience:* we count the number of projects each person participated. The more projects a person worked on, the more experience she gained, and therefore the more capability she could have. Feature values are normalized

using Min-Max normalization and ranged in $[0, 1]$. That is,

$$\text{Normalized}(e) = \frac{e - E_{min}}{E_{max} - E_{min}},$$

where e is the value before normalization, E_{max} is the maximum of value for this feature, and E_{min} is the minimum value.

- *Win Experience:* Only looking at the number of involved projects may not suffice to prove a person's expertise, so we further consider their project outcomes, and count those projects with win outcomes. Feature values are normalized using Min-Max normalization.

- *Win Rate:* We also calculate win rate from each person's experience. It is defined as:

$$\text{Win Rate} = \frac{\text{Number of winning projects}}{\text{Number of participated projects}}.$$

  Note that if number of participated projects is 0, the win rate is also 0.

- *Role Experience:* People recommended for a specific role must have working experience for that role before. It is defined as the number of projects a person worked as that role. This leads to a feature set of a $m \times n$ matrix, where m is the number of roles, and n is the number of all people. This value is also normalized.

5.3.2.2 Team Features From Social Connections

Team features measure how well team members collaborate and communicate with each other. We extract team features from multiple social graphs. First, work collaboration graph is important in determining team's closeness. People who have worked together may collaborate better than people who do not have collaboration experiences. Second, social connections among people may also reflect

their acquaintance or familiarity between each other, e.g., employees of the same division in a company, membership of the same communities, etc. These connections can be depicted as social graphs where nodes are people of interest and edges are their connection relationship.

We demonstrate two team features that are extracted from different social graphs. The first team feature is defined as *team closeness* derived from work collaboration graph. We define a graph $G = (P, E)$ from historical project data, where $P = \{p_1, p_2, \cdots, p_n\}$ represents people, and E represents collaboration relationships from the projects. An edge $e = (p_i, p_j) \in E$, where $p_i, p_j \in P$ exists iff p_i and p_j have worked together on the same project before. A closeness score for a team $T \subset P$ is defined as

$$Closeness = \frac{2}{|T| \times (|T| - 1)} \sum_{p_i, p_j \in T} \frac{1}{ShortestPath(p_i, p_j)},$$

where $|T|$ is the cardinality of team T. The shortest path between two nodes is calculated from the complete work collaboration graph, and we define

$$ShortestPath(p_i, p_j) = |P|$$

if there is no path between p_i and p_j. This feature is adapted from the classical social measure [90]. It characterizes how well an individual belongs to the rest of a team. Note that the higher the score, the closer the teams. The maximum value for a closeness score is 1 which means every person within this team has worked with each other previously.

The second team feature is the connection strength from various social connections. Besides work collaboration history, other connections may also affect the cohesion of teams. In our system, we crawled various connection information from the company intranet, including if same direct manager, internal wiki co-edit relationship, internal instant messenger friendship, and communities co-

membership, etc. We then build a general weighted graph from this connection data, $G_s = (P, E_s)$, where P represents people of interest, and E_s represents various connections between people. An edge $e_{ij} = (p_i, p_j) \in E_s$ iff we find relationships between p_i and p_j. Since there are multiple types of connections between people, we define the weight of e_{ij} as the number of relationships we can find from the crawled data, denoted as $CountPath(p_i, p_j)$. A connection score for team T is calculated as

$$Connection = \frac{2}{|T| \times (|T| - 1)} \sum_{p_i, p_j \in T} CountPath(p_i, p_j).$$

5.3.3 Feature Weight Learning

The ultimate goal for team recommendation is to achieve project success, e.g., winning games in sports, publishing papers in academia, winning deals in business, etc. Besides team strength, there are other factors that may also affect project outcomes in IT strategic outsourcing, e.g., service quality, expected client values, competitor's offerings, etc. However, excluding these factors that are not controlled by a deal team, team strength itself has a significant effect on the project success. Different features have different significance in affecting the project result. Therefore it is important to assign feature weights in a reasonable way. In existing work, feature weights are either not considered or manually assigned. In this work, the feature weights are learned using a machine learning approach by leveraging project outcomes (win and not win). We can also study the correlation between features and the project outcomes during the learning process. We formalize the feature weight learning problem as follows:

Problem 3 (Feature Weight Learning) *Suppose we have a training set of historical projects* $\{S_1, S_2, \cdots, S_t\}$*, teams* $\{T_1, T_2, \cdots, T_t\}$ *which were formed to finish the projects, as well as their project outcomes* $Y = \{y_1, y_2, \cdots, y_t\}$*, where* $y_i \in \{Win, NotWin\}$*. Given the previously defined individual features* F *and team*

features G, we learn their corresponding weight vectors $\overrightarrow{W_1}$ and $\overrightarrow{W_2}$ so that for a set of new projects and candidate teams, we can best predict these teams' project outcomes.

We leverage logistic regression [8] to solve the above weight learning problem. Logistic regression measures correlation between these features and outcomes by learning coefficients of the features, and calculating a probability score to predict the outcome. These coefficients can be treated as weights to measure feature significance. In our case, the project outcome is a binary categorical dependent variable. All the individual and team features are defined as independent variables. For a team T, the learned weight vectors $\overrightarrow{W_1}$ and $\overrightarrow{W_2}$ are then used to compute its team strength score, i.e., $TeamStrength(T)$. Note that we assume all the weights in $TeamStrength(T)$ to be non-negative. We therefore use non-negative logistic regression method [2] to obtain all the weights.

5.3.4 Team Recommendation Algorithm

From the definition of Problem 2, our team recommendation is a combinatorial optimization problem in nature. As pointed out in [47], it is an NP-hard problem. Therefore we need some approximation algorithm to find a quasi optimal solution.

We use a variation of MaxLogit algorithm [79] for finding teams. MaxLogit is an equilibrium selective learning algorithm studied in potential game theory and networking communities. In a potential game that is composed of a set of players and according action sets, every player tries to take the action that will increase a single global utility – the potential function [59]. It is established that every potential game possesses at least one Nash Equilibrium profile that maximizes the potential function [65]. MaxLogit can converge to the best Nash equilibrium with provably fastest convergence rate for the potential games.

[2] `http://cran.r-project.org/web/packages/penalized/`

Our team recommendation problem can be fitted into a potential game too. Consider for each required role, the action set is choosing expert from candidates of that role. The potential function is the combined feature for a team. Our goal is to find the optimal team composition that can maximize the potential function. Therefore the equilibrium selective algorithm such as MaxLogit can be applied here. The MaxLogit algorithm has two major steps: composing a new team solution, and deciding whether to accept the new solution or stick to the old one. The pesudo code of the team search algorithm is presented in Algorithm 1.

Algorithm 1 Finding Best Team

Input: a project with required roles $S' = \{s_1, s_2, \cdots, s_q\}$, set $C(s_i)$ of candidates for each role $s_i \in S'$, individual feature F for each person, team feature G for any team, feature weights $\overrightarrow{W_1}, \overrightarrow{W_2}$, function $Cost(T)$ as the inverse of combined team feature, number of iterations N, smoothing factor τ

Output: the best team T and its cost $Cost(T)$

Randomly select candidate for each role and generate a team T
for $i = 1$ to N **do**
    Calculate $Cost(T)$
    Randomly select a role, and replace it with a randomly selected alternative candidate, get a new team T'
    Calculate $Cost(T')$
    $prob \leftarrow$ PROBABILITY$(Cost(T), Cost(T'))$
    $r \leftarrow random(0, 1)$
    if $r \leq prob$ **then**
        $T \leftarrow T'$
    end if
end for
return $T, Cost(T)$

function PROBABILITY$(Cost(T), Cost(T'))$
    $v_t = exp^{-Cost(T)/\tau}$
    $v_{t'} = exp^{-Cost(T')/\tau}$
    $prob = \frac{v_{t'}}{max(v_t, v_{t'})}$
    return $prob$
end function

Note that we calculate the team cost as the inverse of team strength, and τ is a small positive constant of smoothing factor. This algorithm always searches team

formations that are in proximity of the current one, because we only change the candidate for one role each time. The combined team feature affects the transition probability. That is, when we find a team with higher combined team feature, we will always adopt it. However, when the new team has lower feature value, we are still likely to adopt it. The algorithm is then able to jump out of local optimum and traverse the complete solution space and focus on the *global maximizer* instead.

5.3.5 System Workflow

We show our team recommendation workflow in Figure 5.1. Our system is composed of the following major components:

- **Feature Extractor:** it extracts information from expert profile, historical project data, and work connection records in the company intranet. The extractor outputs individual features and team features.
- **Feature Learner:** it learns weights of individual and team features with respect to project outcomes using logistic regression.
- **Deal Team Recommender:** given a set of required skills or job roles, recommender uses the features extracted from historical data and feature weights learned from feature learner to rank and search optimal teams. A system UI from the recommender present the finalist to users.

5.4 Experiment Results

In this section we present our experimental results. We first demonstrate an application of our team recommendation framework in a real business scenario. Then we evaluate the effectiveness of MaxLogit algorithm in team formation on the public DBLP data set, and compare it with existing work.

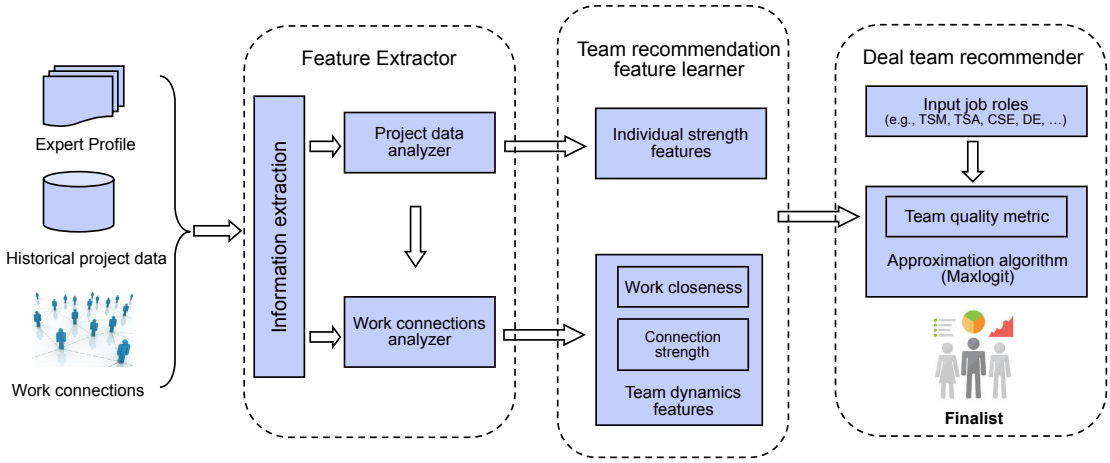


Figure 5.1: Workflow of Team Recommendation System

5.4.1 Team Recommendation in IT Strategic Outsourcing Services

We apply our proposed framework to the team recommendation problem in IT strategic outsourcing services for a large IT service provider. In this subsection, we present the experiment results.

5.4.1.1 Experiment Settings

We collect historical project data in IT strategic outsourcing services within a certain time window. Incomplete records are removed from this data set according to the following two rules. First, some essential roles must be present in each project, e.g. the manager role. Second, each record must have project outcome for our evaluation purpose. Only people who have complete name format are retained. We conduct a few other data cleaning steps, including merging records that have same people but with slightly different name formats. These complete names can be used later for information retrieval from the company intranet, because each employee in the company has a unique name that can be used as identity. We also only retain projects with kick-off date so that we can conveniently sort data records

in a temporal manner. This preprocessing leads to 3,022 project records and 2,142 people as candidates in our data set.

All the records are sorted according to the projects' kick off date. The first 2,700 projects, about 90% of the data set, are used as history in the experiments. We extract all individual features for every candidate from this history data. We also build collaboration graph based on this history data. Then for the left 322 projects, we can retrieve all individual feature and team closeness values.

In addition, the company intranet provides many platforms for employees to build acquaintance with each other. To name a few, there is friend relationship on the instant message platform. The company also provides online community service to employees. People can join diverse communities according to their own personal interests. Such same community membership is also a social connection we consider would improve team collaboration. Therefore we also crawl all such social connection information among employees from the company intranet. In fact, the company provides an API to efficiently access these assorted social connections between any two employees. By submitting pairs of employee identity to the API, we crawl all possible social connections for people in our data set. As a result, 1,413 candidates' connection information is successfully retrieved. This data is used for the team connection feature in our experiments.

5.4.1.2 Feature Validation

We first carry out validation study over the proposed features. We divide our data set, namely the 322 projects data, into "Win" and "NotWin" groups, and examine their feature difference by performing the statistical hypothesis two sample t-test. The test examines whether the Win teams have higher feature means than NotWin teams. The validation result is shown in Table 5.2. Note that all the feature values are normalized in the test.

We find that most features in winning teams have higher mean values than

non-winning teams, especially in individual features. The p-values of the individual features in the test indicate that such differences are quite significant. Team Closeness also shows obvious advantage in winning teams. However, the Connection value do not show similar difference. Instead the Connection difference for team features is quite slight between Win and NotWin teams.

Table 5.2: 2-Sample t-test of features

	Mean of Win Teams	**Mean of NotWin Teams**	**P-value**
Experience	0.4921	0.3805	0.00002
Win Experience	0.4351	0.3243	4.535e-05
Win Rate	0.8352	0.7176	1.966e-08
Role Experience	0.7467	0.5839	1.162e-06
Team Closeness	0.4111	0.3717	0.0317
Connection	0.1896	0.1924	0.4727

5.4.1.3 Feature Weight Learning

Feature weights are learned using Logistic Regression method. The learned feature weights are shown in Table 5.3. This method gives us six non-negative coefficients, excluding win experience feature. Notice that our purpose here is to use feature weights for ultimate team strength calculation, not the classification results per se, therefore the coefficients can also be adjusted as needed accordingly.

Table 5.3: Feature Weight Learning Results from Non-negative Logistic Regression

Feature	**Weight**
Experience	0.1545
Win Rate	2.9949
Role Experience	1.9881
Team Closeness	0.4993
Connection	1.8699
Intercept	-3.6404

5.4.1.4 Team Search Algorithm Evaluation

To evaluate the algorithm effectiveness, we generate 50 random team requirements, each of which requires 3 to 7 randomly selected roles. For comparison purpose, we include two heuristic algorithms as baseline. The first is BestRole algorithm which ignores team features and always chooses the best candidate for each role. The second is TopK traversal algorithm which traverses all possible combinations of the top k candidates and chooses the team with highest team score. Using feature weights learned from previous step, we calculate team strength of solutions generated by each algorithm. In our experiments we set k to 5 in TopK traversal method. For MaxLogit, the smoothing constant τ is set to 0.05, and the iteration number N is 1,000.

The results are illustrated in Figure 5.2 and Figure 5.3. From Figure 5.2 we can see that almost in every case, MaxLogit algorithm can find teams with higher team strength score than TopK and BestRole algorithms. BestRole strategy is almost always the worst among these three methods. This confirms that team feature plays an important role in determining team strength. Without considering the team feature, simply choosing the best individual candidate for each role cannot guarantee the best team. We also compare the average team strength for all generated teams. As demonstrated in Figure 5.3, MaxLogit also achieves the highest average team strength score among the three algorithms.

5.4.2 Experiments over DBLP Data Set

To demonstrate the expandability of our algorithm, we also apply it to the public DBLP data set as the paper [47]. The authors in [47] designed two different team cost functions, and came up with two different algorithms to search the best team accordingly. We apply our algorithm to both scenarios and compare the performance with the two algorithms in this section.

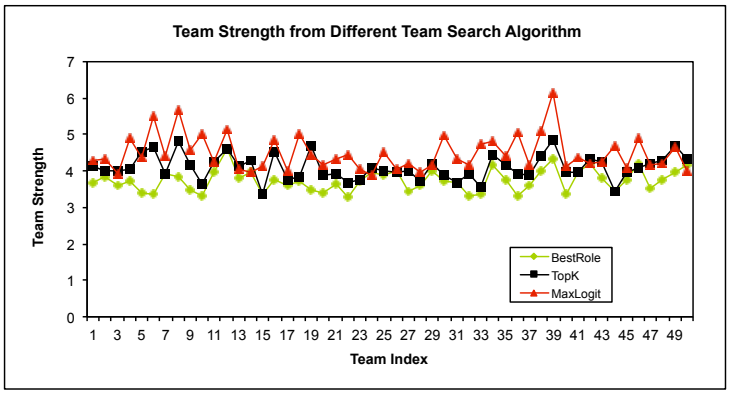


Figure 5.2: Best Team Strength

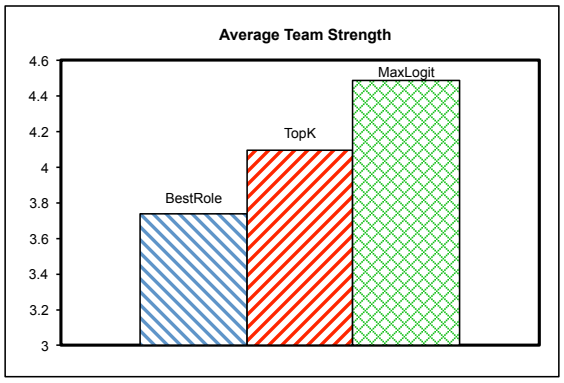


Figure 5.3: Average Team Strength

To make a fair comparison, we use the same setting as [47]. The data set is a snapshot of DBLP data taken on Nov 26, 2013 [3]. This data set contains a collection of papers and their author names in computer science. We generated a set of experts and their skills in the same way as the paper [47]. More specifically, we only keep papers published in some specified major conferences in the areas of Database (DB), Data Mining (DM), Artificial Intelligence (AI), and Theory (T). A total of 19 venues are kept here: DB={SIGMOD, VLDB, ICDE, ICDT, EDBT, PODS}, DM={KDD, WWW, SDM, PKDD, ICDM}, AI={ICML, ECML, COLT, UAI}, and T={SODA, FOCS, STOC, STACS}. The expert set consists of authors who have at least three papers in this dataset. The skills of each expert are

[3] `http://dblp.uni-trier.de/xml/dblp.xml`

generated as the set of terms that appear in at least two publications the expert co-authors. We generate a graph from the coauthor relationship where two experts are connected only if they have written at least two papers together. The weight of an edge connecting person i and j is calculated as 1-$\frac{p_i \cap p_j}{p_i \cup p_j}$ where p_i means the set of papers i has published in this data set. This final graph consists of 8,685 nodes and 13,974 edges.

5.4.2.1 Performance Evaluation

We generate a number of projects as follows. Each project is specified by a set of skills. The number of skills varies from 2, 4, 6, ..., 20. For each configuration, we randomly generate 100 projects and calculate the average result from each algorithm.

We first compare the communication cost between MaxLogit algorithm and the algorithm RarestFirst for the *diameter* communication cost problem. The *diameter* is defined as the largest shortest path between any two members in a team. If the solution produced by either algorithm is not a connected graph, we simply ignore it. The original problem setting is to find the best connected team. For such a purpose, the algorithm RarestFirst will also choose members who may not cover required skills as a connection hub. Our algorithm always only considers candidates for each required skill, and it will only output expert selection of each skill. However, in the final team formation we can include all the people on the shortest path of each pair from the skill covering experts. This will generate a solution with the same diameter communication cost as the original output of our algorithm, and it also meets the problem requirement as [47].

The diameter communication cost result is shown in figure 5.4. We can see from Figure 5.4 that MaxLogit can always find teams that have less communication cost than RarestFirst. However, MaxLogit only considers the cost but not the connectivity, this leads to higher number of disconnected teams as shown in

Figure 5.5.

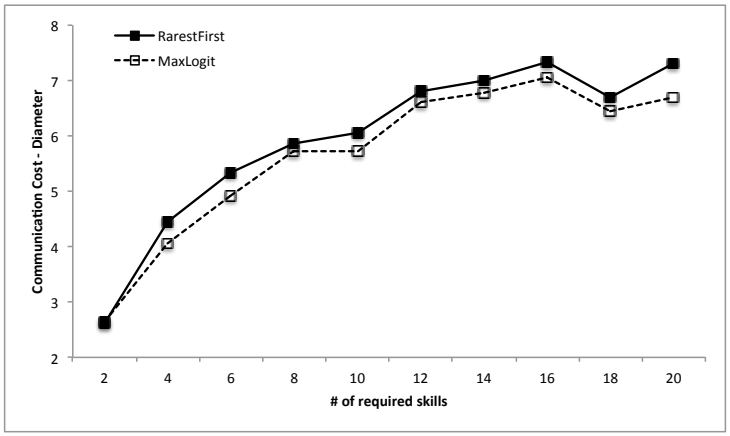


Figure 5.4: Diameter Communication Cost

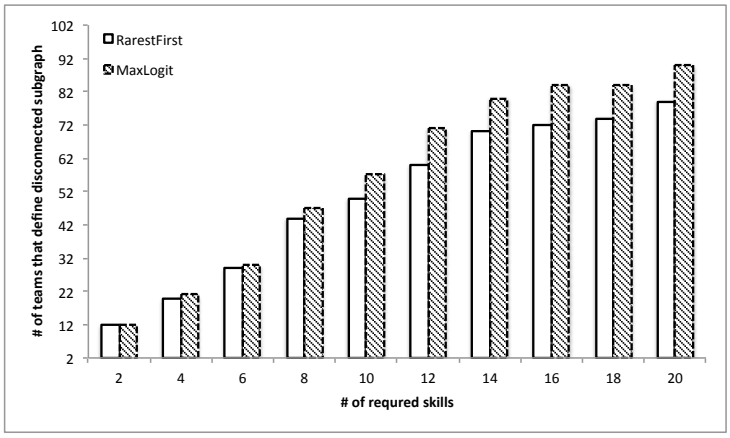


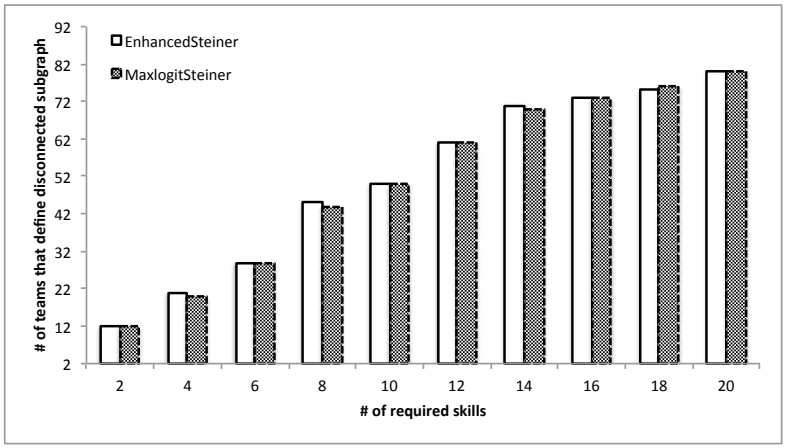
Figure 5.5: # of Disconnected Teams for Diameter Communication Cost Problem

Experiments are also carried out with regard to the communication cost of *minimum spanning tree* using the same team requirements. As shown in Figure 5.6 and Figure 5.7, we can see that MaxLogit can also work well in this setting. However, the communication cost of teams found by MaxLogit is a bit higher than Enhanced Steiner algorithm in [47]. The difference becomes large when it needs a large number of required skills. This is because MaxLogit employs random exploration of team formation. When the number of required skills for a team becomes large, the combination of possible teams also becomes large. While we simply fix the number

of iteration in the MaxLogit algorithm, the proportion of explored solution space becomes relatively smaller. Although our algorithm cannot find as good teams as Steiner Tree algorithm in this setting, we can see the results are still reasonably close.

Figure 5.6: Minimum Spanning Tree cost

Figure 5.7: Number of Disconnected Teams of MST Cost Problem



From the above experiment results, we can see that MaxLogit algorithm works quite well in the two different settings. It can achieve comparable results with the algorithms designed specifically for the two settings. Occasionally MaxLogit's solutions are even better. As we target a general team strength setting, our algorithm demonstrates its expandability in the experiments.

5.5 Conclusions

We study the problem of finding competent deal team for a given project in IT strategic outsourcing services. We leverage the collaboration history and various social networks to measure and evaluate teams' capacity by combining both individual strength and team features. The proposed team recommendation framework is very general and can be applied to other domains as well. Since our framework considers both individual and team level characteristics, previous work that only consider one single type of team communication cost can be easily fitted into our framework as a special case. The feature weights are learned using a machine learning approach by leveraging project outcomes. We score and rank teams by combining all the features in a linear weighted way. We apply an equilibrium selective learning algorithm to find the approximate best team. The proposed framework is applied to the team recommendation problem in a real business scenario. Experimental results show that our framework works well in practice. We also compare our recommendation algorithm with other cost function oriented algorithm on DBLP dataset for team formation in academic paper authoring. The experimental results demonstrate the effectiveness of our algorithm.

In real practice, there are other factors that need to be considered in team recommendation, for example, people's availability. Some team members have strong expertise and can collaborate well with other members, but they are not available. Such availability constraint should be considered in the recommendation. Also, some projects may come at the same time. It is interesting to study how to recommend teams for a set of projects with different priorities. There are a rich set of research topics in team recommendation which can be investigated in our future work.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, we study several data mining problems over user-generated data on the web and social networks. We exploit the results to boost knowledge discovery and recommendation service.

First, we study mining social media data, more specifically Twitter, to discover interesting knowledge. We first explore our investigation about location type detection using individual tweet content. We extend probabilistic text classification models to incorporate temporal features and user history information in terms of probabilistic priors. The experimental results show that our extensions of temporal feature can increase classification accuracy from about 47% to 49% for overall dataset. However, in some specific daily time hours, the improvement is much more significant, e.g., from 37.7% to 45.3% for tweets posted at around 0 o'clock. The extension with users' check-in history leads to a boost in the accuracy from 47.1% to 57% for Maximum Entropy.

Then we study the problem of leveraging Twitter to quantify political legitimacy for specific populaces. We design a framework that converts tweets into a number of topic dimensions using the probabilistic topic modeling. We leverage the sentiment

analysis technique to evaluate polarity of each tweet, and aggregate a large number of tweets into the final legitimacy score of a populace. The empirical evaluation on eight sample countries using related public tweets demonstrates that our proposed framework shows a strong correlation to results reported in political science literature, with the coefficient value of 0.7997887 (P- value = 0.01717). We also apply this framework to a data set collected from traditional news media, and compare it with social media Twitter. The results reveal that social media may work differently than traditional news media for such political legitimacy quantification task.

Moreover, we investigate how to exploit user generated data to improve recommendation services. In particular, we present our work of mining implicit user feedback in a cold-start problem of video recommendation. We propose a classification framework based on previously seen video pairs, and learn the weights of video attributes for ranking candidate videos to recommend. This framework leads to a mean average R-precision score of 25%, compared to the baseline of 21% without co-view information.

Furthermore, social networks can also be exploited to improve recommendation tasks. We demonstrate our work by studying the problem of team recommendation. To quantitatively capture the team level features, we take various social networks among people into consideration from project history and many other online activities. Moreover, we learn the feature weights from the training dataset based on the correlation between features and project outcomes. We apply a combinatorial optimization algorithm to search the approximate best team. We validate our approach experimentally in a real business scenario and also compare our approach with other state-of-the-art methods using public DBLP dataset. The results demonstrate the effectiveness of our approach.

6.2 Future Work

Many directions are worth further pursuing in future. For the social media data, we believe there are various ways to further improve understanding. For example, for the task of classifying users' location type using tweet, we can further investigate recurrent patterns, and utilize such temporal advantage to improve classification accuracy. Moreover, social media are always evolving, there will always be new opportunities to discover interesting knowledge.

Another direction to improve leveraging user-generated data is to combine multiple formats of data for recommendation service. In this thesis, we only consider limited data formats like implicit feedback and social networks for recommendation tasks. There may be social platforms that can harvest various formats of data, including text, image, social network relationships, etc. By integrating techniques of text mining and social network analysis, we can address many other challenges and opportunities to improve the recommendation services.

Bibliography

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE TKDE*, 2005.

[2] N. Antulov-Fantulin, M. Bošnjak, T. Šmuc, M. Jermol, M. Žnidaršič, M. Grčar, P. Keše, and N. Lavrač. Ecml/pkdd 2011 - discovery challenge: "videolectures.net recommender system challenge". *http://tunedit.org/challenge/VLNetChallenge*, 2011.

[3] G. R. Arce. *Nonlinear Signal Processing: A Statistical Approach*. Wiley, 2005.

[4] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[5] A. Bakliwal, J. Foster, J. van der Puil, R. O'Brien, L. Tounsi, and M. Hughes. Sentiment analysis of political tweets: towards an accurate classifier. Association for Computational Linguistics, 2013.

[6] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In *WWW*, 2008.

[7] A. Baykasoglu, T. Dereli, and S. Das. Project team selection using fuzzy optimization approach. *Cybern.Syst.*, 38(2):155–185, 2007.

[8] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *The Journal of machine Learning research*, 3:993–1022, 2003.

[10] V. Bobicev and M. Sokolova. An effective and robust method for short text classification. In *Proceedings of the 23rd national conference on Artificial intelligence - Volume 3*, AAAI'08, pages 1444–1445. AAAI Press, 2008.

[11] J. Bollen, H. Mao, and A. Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *ICWSM*, 2011.

[12] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *J. of Computational Science*, 2(1):1–8, 2011.

[13] S. Braghin, J. Yong, A. Ventresque, and A. Datta. SWAT: social web application for team recommendation. In *2012 IEEE 18th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 845–850, 2012.

[14] P. Brockwell and R. Davis. *Time Series: Theory and Methods*. Springer Series in Statistics. Springer, 2009.

[15] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 1998.

[16] H.-W. Chang, D. Lee, M. Eltaher, and J. Lee. @phillies tweeting from philly? predicting twitter user locations with spatial word usage. In *IEEE/ACM ASONAM*, 2012.

[17] S.-J. Chen and L. Lin. Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering. *IEEE Transactions on Engineering Management*, 51(2):111–124, 2004.

[18] Z. Cheng, J. Caverlee, K. Y. Kamath, and K. Lee. Toward traffic-driven location-based web search. In *ACM CIKM*, 2011.

[19] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *ACM CIKM*, 2010.

[20] F. C. T. Chua, W. W. Cohen, J. Betteridge, and E.-P. Lim. Community-based classification of noun phrases in twitter. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1702–1706, New York, NY, USA, 2012. ACM.

[21] N. Dalvi, R. Kumar, and B. Pang. Object matching in tweets with spatial models. In *ACM WSDM*, 2012.

[22] A. Datta, S. Braghin, and J. T. T. Yong. The zen of multidisciplinary team recommendation. arXiv e-print 1303.0646, School of Computer Engineering Nanyang Technological University, Singapore, Mar 2013.

[23] A. Datta, J. Tan Teck Yong, and A. Ventresque. T-RecS: team recommendation system through expertise and cohesiveness. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, pages 201–204, New York, NY, USA, 2011. ACM.

[24] T. De Smedt and W. Daelemans. Pattern for python. *Journal of Machine Learning Research*, 13:2063–2067, 2012.

[25] S. Debnath, N. Ganguly, and P. Mitra. Feature weighting in content based recommendation system using social network analysis. In *WWW*, 2008.

[26] D. Ediger, K. Jiang, J. Riedy, D. Bader, C. Corley, R. Farber, and W. Reynolds. Massive social network analysis: Mining twitter for social good. In *2010 39th International Conference on Parallel Processing (ICPP)*, pages 583–593, Sept. 2010.

[27] J. Evans, S. Fast, and N. Markuzon. Modeling the social response to a disease outbreak. In *SBP*, pages 154–163, 2013.

[28] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37, 1996.

[29] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme. Learning Attribute-to-Feature mappings for Cold-Start recommendations. In *ICDM*, 2010.

[30] B. Gilley. The meaning and measure of state legitimacy: Results for 72 countries. *European J. of Political Research*, 45(3):499–525, 2006.

[31] B. Gilley. State legitimacy: An updated dataset for 52 countries. *European J. of Political Research*, 51(5):693–699, 2012.

[32] P. Gonçalves, M. Araújo, F. Benevenuto, and M. Cha. Comparing and combining sentiment analysis methods. In *Proceedings of the First ACM Conference on Online Social Networks*, COSN '13, pages 27–38, New York, NY, USA, 2013. ACM.

[33] J. Han and M. Kamber. *Data Mining: Concepts and Techniques, 2nd ed.* Morgan Kaufmann Publishers, 2006.

[34] B. Hecht, L. Hong, B. Suh, and E. H. Chi. Tweets from justin bieber's heart: The dynamics of the location field in user profiles. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 237–246, New York, NY, USA, 2011. ACM.

[35] R. S. Huckman, B. R. Staats, and D. M. Upton. Team familiarity, role experience, and performance: Evidence from indian software services. *Manage. Science*, 55(1):85–100, Jan. 2009.

[36] C. Hutto and E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[37] O. Jin, N. N. Liu, K. Zhao, Y. Yu, and Q. Yang. Transferring topical knowledge from auxiliary long texts for short text clustering. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 775–784, New York, NY, USA, 2011. ACM.

[38] T. Joachims. Making large-scale support vector machine learning practical. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods*, pages 169–184. MIT Press, Cambridge, MA, USA, 1999.

[39] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

[40] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2000.

[41] M. Kargar and A. An. Discovering top-k teams of experts with/without a leader in social networks. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 985–994, New York, NY, USA, 2011. ACM.

[42] M. Kargar, A. An, and M. Zihayat. Efficient bi-objective team formation in social networks. In *Proceedings of the 2012 European conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II*, ECML PKDD'12, pages 483–498, Berlin, Heidelberg, 2012. Springer-Verlag.

[43] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*, 2008.

[44] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[45] G. Laboreiro, L. Sarmento, J. Teixeira, and E. Oliveira. Tokenizing micro-blogging messages using a text classification approach. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, AND '10, pages 81–88, New York, NY, USA, 2010. ACM.

[46] A. Lamb, M. J. Paul, and M. Dredze. Investigating twitter as a source for studying behavioral responses to epidemics. In *AAAI Fall Symp.*, 2012.

[47] T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 467–476, New York, NY, USA, 2009. ACM.

[48] K. Leetaru and P. Schrodt. GDELT: global database of events, language, and tone, 1979-2012. In *International Studies Association Annual Conference*, San Francisco, CA, USA, April 2013.

[49] W. Li, P. Serdyukov, A. P. de Vries, C. Eickhoff, and M. Larson. The where in the tweet. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 2473–2476, New York, NY, USA, 2011. ACM.

[50] C.-Y. Lin, N. Cao, S. Liu, S. Papadimitriou, J. Sun, and X. Yan. SmallBlue: social network analysis for expertise search and collective intelligence. In *Data Engineering, 2009. ICDE '09. IEEE 25th International Conference on*, pages 1483–1486, 2009.

[51] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 2003.

[52] H. Liu, S. Das, D. Lee, P. Mitra, and C. L. Giles. Using co-views information to learn lecture recommendations. *Proc. of ECML-PKDD 2011 Discovery Challenge Workshop*, pages 71–82, 2011.

[53] H. Liu and D. Lee. Quantifying political legitimacy from twitter. In W. G. Kennedy, N. Agarwal, and S. J. Yang, editors, *Social Computing, Behavioral-Cultural Modeling and Prediction*, number 8393 in Lecture Notes in Computer Science, pages 111–118. Springer International Publishing, Jan. 2014.

[54] H. Liu, B. Luo, and D. Lee. Location type classification using tweet content. In *Proceedings of the 2012 11th International Conference on Machine Learning and Applications - Volume 01*, ICMLA '12, pages 232–237, Washington, DC, USA, 2012. IEEE Computer Society.

[55] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pages 186–193, New York, NY, USA, 2004. ACM.

[56] Z. Liu, W. Yu, W. Chen, S. Wang, and F. Wu. Short text feature selection for micro-blog mining. In *2010 International Conference on Computational Intelligence and Software Engineering (CiSE)*, pages 1 –4, Dec. 2010.

[57] C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[58] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[59] J. Marden, G. Arslan, and J. Shamma. Cooperative control and potential games. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(6):1393–1407, 2009.

[60] A. K. McCallum. MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu, 2002.

[61] Q. Mei, D. Zhang, and C. Zhai. A general optimization framework for smoothing language models on graph structures. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'08, pages 611–618, New York, NY, USA, 2008. ACM.

[62] P. Melville and V. Sindhwani. Recommender systems. In *Encyclopidia of Machine Learning*. Springer, 2010.

[63] Mika Timonen. Categorization of very short documents. In *Proceedings of the 4th International Conference on Knowledge Discovery and Information Retrieval*, KDIR 2012, Barcelona, Spain, Oct. 2012. Springer.

[64] M. Moens, J. Li, and T. Chua. *Mining User Generated Content*. Social Media and Social Computing. Taylor & Francis, 2014.

[65] D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124 – 143, 1996.

[66] K. Nigam. Using maximum entropy for text classification. In *In IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.

[67] K. Nishida, T. Hoshide, and K. Fujimura. Improving tweet stream classification by detecting changes in word probability. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 971–980, New York, NY, USA, 2012. ACM.

[68] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *ICWSM*, 2010.

[69] M. J. Paul and M. Dredze. You are what you tweet: Analyzing twitter for public health. In *ICWSM*, 2011.

[70] M. J. Paul and M. Dredze. A model for mining public health topics from twitter. *Health*, 11:6–16, 2012.

[71] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, volume 4321, pages 325–341. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[72] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 91–100, New York, NY, USA, 2008. ACM.

[73] Q. Pu and G.-W. Yang. Short-text classification based on ICA and LSA. In *Proceedings of the Third international conference on Advnaces in Neural Networks - Volume Part II*, ISNN'06, pages 265–270, Berlin, Heidelberg, 2006. Springer-Verlag.

[74] D. Rao, D. Yarowsky, A. Shreevats, and M. Gupta. Classifying latent user attributes in twitter. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, pages 37–44. ACM, 2010.

[75] J. Rodgers and A. Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988.

[76] H. Saif, Y. He, and H. Alani. Semantic sentiment analysis of twitter. In *The Semantic Web–ISWC 2012*, pages 508–524. Springer, 2012.

[77] J. J. Salerno, B. Romano, and W. Geiler. The national operational environment model (noem). In *SPIE Modeling and Simulation for Defense Systems & App.*, 2011.

[78] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. P. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 2013.

[79] Y. Song, S. H. Wong, and K.-W. Lee. Optimal gateway selection in multi-domain wireless networks: a potential game perspective. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, MobiCom '11, pages 325–336, New York, NY, USA, 2011. ACM.

[80] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 841–842, New York, NY, USA, 2010. ACM.

[81] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in twitter to improve information filtering. In *ACM SIGIR*, pages 841–842, 2010.

[82] T. Strohman, W. B. Croft, and D. Jensen. Recommending citations for academic papers. In *SIGIR*, SIGIR '07, 2007.

[83] A. Sun. Short text classification using very few words. In *Proc. of ACM SIGIR Conference (SIGIR'12)*, 2012.

[84] X. Sun, H. Wang, and Y. Yu. Towards effective short text deep classification. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 1143–1144, New York, NY, USA, 2011. ACM.

[85] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *ICWSM*, pages 178–185, 2010.

[86] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Election forecasts with twitter how 140 characters reflect the political landscape. *Social Science Computer Review*, 29(4):402–418, 2011.

[87] A. H. Turpin and W. Hersh. Why batch and user evaluations do not give the same results. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 225–231, New York, NY, USA, 2001. ACM.

[88] A. Ventresque, J. T. T. Yong, and A. Datta. Impact of expertise, social cohesiveness and team repetition for academic team recommendation. In *Social Informatics*, volume 6984, pages 296–299. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[89] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan. A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In *Proceedings of the ACL 2012 System Demonstrations*, pages 115–120. Association for Computational Linguistics, 2012.

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

[91] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok. Interpreting TF-IDF term weights as making relevance decisions. *ACM Trans. Inf. Syst.*, 26(3):13:1–13:37, June 2008.

[92] M. Ye, K. Janowicz, C. MÃijlligann, and W.-C. Lee. What you are is when you are: the temporal dimension of feature types in location-based social networks. In *ACM SIGSPATIAL*, 2011.

[93] Q. Yuan, G. Cong, and N. M. Thalmann. Enhancing naive bayes with various smoothing methods for short text classification. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 645–646, New York, NY, USA, 2012. ACM.

[94] S. Zelikovitz. Transductive LSI for short text classification problems. In V. Barr and Z. Markov, editors, *FLAIRS Conference*. AAAI Press, 2004.

[95] S. Zelikovitz and H. Hirsh. Improving short text classification using unlabeled background knowledge to assess document similarity. In *In Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1183–1190, 2000.

[96] S. Zelikovitz and H. Hirsh. Improving short-text classification using unlabeled data for classification problems. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 1191–1198, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[97] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, Apr. 2004.

[98] A. Zzkarian and A. Kusiak. Forming teams: an analytical approach. *IIE Transactions*, 31:85–97, 2004.

Vita

Haibin Liu

EDUCATION

The Pennsylvania State University, University Park, PA 08/2007 – 08/2014
Ph.D. in Information Sciences and Technology

Peking University, Beijing, China 09/2004 – 07/2007
M.S. in Dept. of Intelligence Science, School of EECS

Peking University, Beijing, China 09/2000 – 07/2004
B.S. in Computer Science and Technology

EXPERIENCE

IBM Research Almaden Intern, San Jose, CA 05/2013 – 08/2013
IBM T.J. Watson Research Center Intern, Hawthorne, NY 05/2008 – 08/2008

PUBLICATIONS

Haibin Liu, Dongwon Lee: "Quantifying Political Legitimacy from Twitter", Int'l Conf. on Social Computing, Behavioral Modeling, and Prediction (SBP), Washington DC, USA, April 2014

Haibin Liu, Bo Luo, Dongwon Lee: "Location Type Classification Using Tweet Content", 11th IEEE Int'l Conf. on Machine Learning and Applications (ICMLA), Boca Raton, FL, USA, December 2012

Haibin Liu, Woo-Cheol Kim, Dongwon Lee: "Characterizing Landing Pages in Sponsored Search", 8th Latin American Web Congress (LA-WEB), Cartagena, Colombia, October 2012

Haibin Liu, Sujatha Das Gollapalli, Dongwon Lee, Prasenjit Miltra, C.Lee Giles: "Using Co-view Information to Learn Lecture REdommendations", ECML/PKDD 2011 – Discovery Challenge Workshop, Athens, Greece, Sept 2011 (8^{th} place out of 62 teams for track1, 4^{th} place out of 22 teams for track 2)

AWARDS AND HONORS

Second prize of "IBM CUP" Campus Creation Design Competition (group member, group finished 3^{rd} out of 265 participants), 2005

Excellent Final Thesis of Bachelor Degree, 2004