Mining Content and Relations for Social Spammer Detection

by

Xia Hu

A Dissertation Presented in Partial Fulfillment
of the Requirement for the Degree
Doctor of Philosophy

Approved April 2015 by the
Graduate Supervisory Committee:

Huan Liu, Chair
Subbarao Kambhampati
Jieping Ye
Christos Faloutsos

ARIZONA STATE UNIVERSITY

May 2015

ABSTRACT

Social networking services, like Facebook and Twitter, have emerged as an important platform for large-scale information sharing and communication. With the growing popularity of social media, spamming has become rampant in the platforms. Many (fake) accounts, known as spammers, are employed to overwhelm other users with unwanted information in social media. Complex network interactions and evolving content present great challenges for social spammer detection. Different from some existing well-studied platforms, distinct characteristics of newly emerged social media data present new challenges for social spammer detection. First, texts in social media are short and potentially linked with each other via user connections. Second, it is observed that abundant contextual information may play an important role in distinguishing social spammers and normal users. Third, not only the content information but also the social connections in social media evolve very fast. Fourth, it is easy to amass vast quantities of unlabeled data in social media, but would be costly to obtain labels, which are essential for many supervised algorithms. To tackle those challenges raise in social media data, I focused on developing effective and efficient machine learning algorithms for social spammer detection.

I provide a novel and systematic study of social spammer detection in the dissertation. By analyzing the properties of social network and content information, I propose a unified framework for social spammer detection by collectively using the two types of information in social media. Motivated by psychological findings in physical world, I investigate whether sentiment analysis can help spammer detection in online social media. In particular, I conduct an exploratory study to analyze the sentiment differences between spammers and normal users; and present a novel method to incorporate sentiment information into social spammer detection framework. Given the rapidly evolving nature, I propose a novel framework to efficiently reflect the effect of newly emerging social spammers. To tackle the problem of lack of labeling data in social media, I study how to incorporate network information into text content modeling, and design strategies to select the most representative and informative instances from social media for labeling. Motivated by publicly available label information from other media platforms, I propose to make use of knowledge learned from cross-media to help spammer detection on social media.

To my parents and wife for their love and support

ACKNOWLEDGEMENTS

I would like to thank my advisor, Huan Liu, for his guidance, encouragement, and support during my dissertation research. He is an outstanding mentor, an easygoing friend, and the most dedicated researcher I have ever known. The experiences with him are my lifelong assets. I would like to thank my dissertation committee members, Subbarao Kambhampati, Jieping Ye and Christos Faloutsos, for their valuable interactions and feedback.

I have been working closely with many colleagues from ASU and outside ASU during my Ph.D. I want to thank Yi Chang, Yan Chen, Tat-Seng Chua, Xiaodong Fan, Biyun Hu, Yan Ke, Zhoujun Li, Yafeng Lu, Hao Ma, Ross Maciejewski, Raymond Migrino, Zhaoyan Ming, Nan Sun, Hari Sundaram, Jun Wang, Kai Wang, Senzhang Wang, Zheng Yang, Philip S. Yu, Chao Zhang, Xiaoming Zhang, Yanchao Zhang.

Members of our Data Mining and Machine Learning Lab inspired me a lot through discussions, seminars, and project collaborations, and I would like to thank the following people for their valuable interactions: Ali Abbasi, Salem Alelyani, Geoffrey Barbier, Huiji Gao, Pritam Gundecha, Isaac Jones, Shamanth Kumar, Jundong Li, Fred Morstatter, Tahora Hossein Nazer, Suhas Ranganath, Jiliang Tang, Lei Tang, Robert Trevino, Ran Wang, Suhang Wang, Xufei Wang, Liang Wu, Reza Zafarani.

Over the years, many friends and colleagues have provided me very helpful suggestions, insightful comments, support and encouragement during my Ph.D study. I would like to thank James Caverlee, Jianhui Chen, Jing Gao, Quanquan Gu, Jingrui He, Yuheng Hu, Shuiwang Ji, Ee-Peng Lim, Lydia Manikonda, Aixin Sun, Kartik Talamadupula, Hanghang Tong, Fei Wang, Yilin Wang, Qiang Zhang, Jiayu Zhou.

This material is based upon work supported by, or in part by, the U.S. Army Research Office (ARO) under contract/grant number 025071, the National Science Foundation (NSF) under grant number IIS-1217466, and the Office of Naval Research (ONR) under grant number N000141010091.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Chapter 1

INTRODUCTION

Social media services, such as Facebook, Twitter and Instagram, are increasingly used for individuals to interact with their friends and industries conduct business. Internet users are transforming from information consumers to producers by using social media to easily and collaboratively create content. Social media has emerged as an important platform for large-scale information sharing and communication in various scenarios such as marketing, journalism or public relations.

With the increasing popularity of social media services, social spamming has become rampant. Social spam is defined as “unwanted spam content appearing on social networks and any website with user-generated content (comments, chat, etc.). It can be manifested in many ways, including bulk messages, profanity, insults, hate speech, malicious links, fraudulent reviews, fake friends, and personally identifiable information[1]”. Many fake accounts, known as social spammers [101], are employed to unfairly overpower normal users. A recent example of social spamming reported by Symantec[2] is that Twitter spammers target NFL and MIley Cyrus fans. Many fans of Denver Broncos and Seattle Seahawks have been subjected to a torrent of Twitter spammers. Also, fans of pop star Miley Cyrus have also been plagued with an identical spam campaign on Twitter. The spamming consists three steps: (1) Twitter spammers reply to other users with photo attachments that claim to offer prizes related to the NFL or Miley Cyrus; (2) By directing users to some scam websites, the websites require the users to verify Twitter usernames; (3) Users are asked to participate in some online events such as completing a survey or downloading mobile apps. By conduct-

[1] http://en.wikipedia.org/wiki/Social_spam

[2] http://www.symantec.com/connect/blogs/twitter-spam-bots-target-nfl-and-miley-cyrus-fans

ing the three steps, social spammers can launch various attacks such as befriending victims and then grabbing their personal information [5], conducting spam campaigns which lead to phishing, malware, and scams [37], and conducting political astroturf [85, 86]. Successful spammer detection in social media presents its significance to improve the quality of user experience, and to positively impact the overall value of the social systems going forward [63]. To this end, I am motivated to take advantage of data mining methods to better understand social spammers and improve the detection performance. However, different from the existing intensively studied platforms, such as emails [7], SMS [36] and the web [100], characteristics of newly emerged social media data present new challenges for the task of spammer detection:

First, content analysis for spammer detection in social media has been little studied due to the distinct features of social media messages that are short, unstructured and potentially networked with each other. For example, Twitter allows users to post messages up to 140 characters [59]. Short messages bring new challenges to traditional text analytics. They cannot provide sufficient context information for effective similarity measure, the basis of many text processing methods [46]. In addition, when composing a message, users often prefer to use newly created abbreviations or acronyms that seldom appear in conventional text documents. For example, messages like “How r u?” and “Good 9t” are popular in social media, but they are not even formal words. Although they provide a better user experience, unstructured expressions make it very difficult to accurately identify the semantic meanings of these messages. Last, social media messages are *networked* [110] in the sense that they are generated by users following some others in social media systems. The traditional assumption in many applications that data instances are independent and identically distributed (i.i.d.) is thus no longer valid for networked messages. The distinct features make traditional text analytics less applicable in social media platforms.

Second, many social media systems [108] like Twitter feature unidirectional user binding, meaning anyone can follow anyone else without prior consent from the followee.[3] Many users simply follow back when they are followed by someone for the sake of courtesy [102]. Due to the *reflexive reciprocity* [49], it is easier for spammers to imitate normal users in social media by quickly accumulating a large number of social relations. A recent study [34] shows that spammers can successfully acquire a number of normal followers, especially those referred to as social capitalists who tend to increase their social capital by following back anyone who follows them. Thus, traditional method which is built upon the assumption that spammers cannot establish an arbitrarily large number of social trust relations with normal users becomes less effective in the task of social spammer detection.

Third, in addition to textual and social network information on social media, it has been observed that abundant contextual information is available. For example, in psychology and social sciences, it is well-established that *microexpressions* [40] play a distinct role in detecting deception. Ekman [29] reported that facial and emotional "microexpressions" could be useful to assist in lie detection after testing a total of 20,000 people [30] from all walks of life. Also, as pointed out by Matsumoto *et al.* [74], one may not conclude that someone is lying if a microexpression is detected but that there is more to the story than is being told. The contextual information might be very important, but how to take advantage of the contextual information for social spammer detection is still an open problem.

Fourth, traditional spammer detection methods become less effective due to the fast evolution of social spammers. Social spammers show dynamic content patterns in social media. Spammers' content information changes too fast to be detected by a static anti-spamming system based on offline modeling [110]. Spammers continue to change their spamming strategies and pretend to be normal users to fool the system. A built system may

[3] Although there is often an option for a user to manually (dis)approve a following request, it is rarely used by normal users for convenience.

become less effective when the spammers create many new, evasive accounts. In addition, many social media sites like Twitter have become a target of link farming [34]. The reflexive reciprocity [102, 49] indicates that many users simply follow back when they are followed by someone for the sake of courtesy. With the perceived social influence, they can avoid being detected by network-based methods. Similar results targeting Renren [105] and Facebook [11] have been reported in literature as well. Existing systems rely on building a new model to capture newly emerging content-based and network-based patterns of social spammers. Given the rapidly evolving nature, it is necessary to have a framework that efficiently reflects the effect of newly emerging data in social spammer detection.

Fifth, labeling information of spammers and normal users is time consuming and labor intensive to obtain in social media. Most of existing work formulates spammer detection as a supervised learning problem. Supervised learning methods aim to learn a model based on training data, which involves a basic assumption that a sufficiently large number of labeled instances are available. In the problem of social spammer detection, labeled data is needed to train a supervised model to determine whether a given user is normal user or spammer. However, labels can be expensive and time consuming to obtain in social media. The lack of labeling data presents great challenges to the application of supervised learning algorithms on social spammer detection.

Social media services often provide abundant information which could be potentially useful for social spammer detection. For example, in Figure 1.1, I depict two types of data available in social media. Left part of Figure 1.1 shows an illustration of social media data which consists of five users and nine messages posted by the users. The five users are connected with each other and user u_4 is a spammer user. As shown in the top right corner of Figure 1.1, the messages can be represented in the form of a message-feature matrix. An intuitive method is to employ content analysis for detecting spammers in social media. Profile-based features [63] such as content and posting patterns are extracted to build

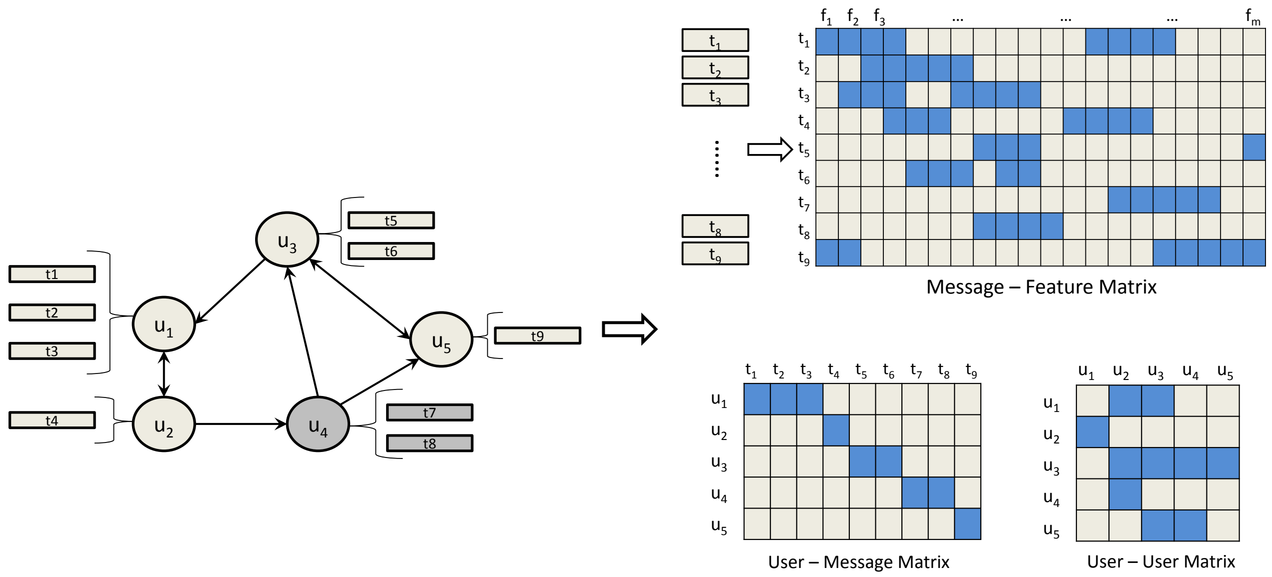


Figure 1.1: Data Representation of Content and Relations in Social Media

an effective supervised learning model, and the model is applied on unseen data to filter social spammers. As shown in the bottom right corner of Figure 1.1, a distinct feature of social media data is that they are potentially networked through user connections, which may contain useful semantic clues that are not available in purely text-based methods. Besides content information, relations between users and messages can be represented via a user-message matrix and a user-user interaction matrix. A possible method is to perform spammer detection by utilizing the social network information [13, 23]. A widely used assumption in the methods is that spammers cannot establish an arbitrarily large number of social trust relations with legitimate users. The users with relatively low social influence or social status in the network will be determined as spammers. Due to the distinct characteristics of social media data, traditional data mining methods become less effective. It motivates us to explore the problem of spammer detection by mining content and relations from new perspectives.

In the dissertation, I study the problem of spammer detection by mining content and relations in social media. Specifically, I investigate the following questions:

- How to model network information and content information seamlessly for the problem of spammer detection in social media?

- How to verify the usefulness of contextual information, and model contextual information for social spammer detection?

- How do we update the built model to efficiently incorporate newly emerging data objects for spammer detection in social media?

- How can we tackle the labeling bottleneck in social media?

By answering the above questions, the main contribution of the dissertation can be summarized as follows. Social spammer detection is a novel and practical problem. In this dissertation, we firstly provide a systematic study from a data mining perspective to understand the characteristics and the challenges of the data. Motivated by data analytics observations, existing social theories and psychological findings, we abstract patterns from social media data which could be useful for the problem of detecting social spammers. Thus we develop statistical learning algorithms for social spammer detection, and achieve good performance comparing to the state-of-the-art methods.

The remainder of this dissertation is structured as follows. In Chapter 2, I review the related work. In Chapter 3, I discuss the proposed unified model of heterogeneous data analytics for social spammer detection. In Chapter 4, I propose a framework for modeling and integrating contextual information. In Chapter 5, I present an online learning scheme to incrementally update the built model. In Chapter 6, I introduce two strategies for active learning. In Chapter 7, I propose a framework to learn knowledge from cross-media resources. In Chapter 8, I conclude and present the future work.

Chapter 2

RELATED WORK

Social spammer detection is a novel and practical problem. Recently, due to the increasing popularity of social media services, it attracts a lot of attention from academia and industry. In this dissertation, I firstly provide a systematic and in-depth study to tackle the problem by developing learning algorithms. There are several lines of related work.

(1) Spammer Detecion

Spammer detection on various platforms, e.g., email [7], SMS [36] and the Web [100], have been studied for years. The spams are designed to corrupt the user experience by spreading ads or driving traffic to particular web sites [100]. A popular and well-developed approach for anti-spam applications is learning-based filtering. The basic idea is that we extract effective features from the labeled data and build a classifier. We then classify new users / messages as either spam or ham according to their content information.

(2) Spammer Detection in Social Media

With the popularity of social media services, there are significant efforts to detect and analyze spammers in various social media sites, including Facebook [11], Twitter [34], Renren [105], etc. Following the efforts of spammer detection in other platforms, some work [63] has been done to study characteristics related to tweet content and user behavior for spammer detection in social media. By understanding spammer activities in social networks, features are extracted to perform effective spammer detection. However, the behaviors of the spammers in social media change too fast to be detected by a traditional anti-spamming system that is based on extensive offline feature building [110]. Since the spammers always create new and evasive patterns to fool the systems, a rule-based system that works well in detecting existing spammers may fail to do so very soon.

Another effective way to perform social spammer detection is to utilize the social network information [13, 23]. A widely used assumption is that spammers cannot establish an arbitrarily large number of social trust relations with normal users. This assumption might not hold in many social networks. Yang *et al.* [105] studied the spammers in Renren, the largest OSN in China similar in features to Facebook. Their results reveal that spammers on Renren can have their friend requests accepted by many normal users and thus well blend into the Renren social graph. A similar result targeting Facebook is reported in [11], where the term "social bots" instead of spammers is used. In contrast to Facebook-like OSNs, microblogging systems feature unidirectional user bindings because anyone can follow anyone else without prior consent from the followee. Ghosh *et al.* [34] show that spammers can successfully acquire a number of normal followers, especially those referred to as social capitalists who tend to increase their social capital by following back anyone following them. Some methods [49, 110] have also proposed to collectively use content and social network information in social spammer detection.

(3) Sentiment Analysis in Social Media.

Sentiment analysis on product reviews has been a hot topic for quite a few years [67]. Recently, the opinion-rich resources in social media attracted attention from disciplines. As an effective tool to understand opinions of the public, sentiment analysis is widely applied in various social media applications [47], including poll rating prediction [80], event prediction [9], etc. O'Connor *et al.* [80] found strong correlation between the aggregated sentiment and the manually collected poll ratings. Bollen *et al.* [10] proposed to measure the dynamic sentiments on Twitter, and compared the correlation between public sentiments and major events, including the stock market, crude oil prices, elections and Thanksgiving. Motivated by the applications of sentiment analysis and the psychological theories, I investigate the use of sentiment information for social spammer detection in this dissertation.

(4) Opinion Spam Detection

It is popular for people to read opinions for various purposes, such as buying a product or visiting a restaurant. Positive opinions can lead to significant financial gains and/or fames for organizations and individuals. This gives good incentives for opinion spam [57]. Opinion spam detection is an important research topic in sentiment analysis and opinion mining [67]. The objective of this task is to detect spam activities in comments about news articles, blogs, or reviews about products or movies. Our studied problem is different from opinion spam detection. First, I aim to examine spam users in stead of spam review texts, which are often assumed to be independent and identically distributed (i.i.d.). Second, I study a general social spammer detection problem, while opinion spams are always topic-oriented.

(5) Cross-Media Learning

Some efforts have been made to employ domain adaption and transfer learning in various applications, e.g. sentiment analysis [66] and text classification [83]. Our work started the investigation of leveraging knowledge from other media for spammer detection in microblogging. Different from traditional methods, based on the quantitatively linguistic variation analysis, our proposed framework naturally combines knowledge learned from internal and external data sources in a unified model. In addition, some work has been done to study the linguistic challenges of social media texts. It is accepted that texts in social media are noisy, but it is also reported by researchers that the texts are not as noisy as what people expected [3]. The language used in Twitter is more like a projection of the language of formal media like news and blogs with shorter form [52], and it is possible to make use of normalization and domain adaption to “clean” it [28]. The evidence provided by linguists also motivate us to explore the language differences of spams across different media, and make use of resources from other media to help spammer detection in microblogging.

(6) Active Learning

As an effective way to tackle the labeling bottleneck, active learning has been extensively studied in various domains for years. Existing methods focus on the data represented by feature vectors [87], and they can be generally categorized into three groups. First, active learners select either the most uncertain instances determined by a single classifier [2, 96] or a committee of classifiers [22, 31]. These approaches always evaluate the data instances separately, thus can not utilize the structure of the data. The second group of methods exploit cluster structure in data, and select instances in each cluster to avoid sampling bias [25, 79, 103]. The key idea of these approaches is to identify a sophisticated cluster structure based on content information. The key limitation of these methods is that they cannot well utilize information from labeled data. Different from traditional approaches, our proposed framework incorporates relation information into the content modeling, and further selects instances by taking advantage of the social network structure.

Chapter 3

HETEROGENEOUS INFORMATION ANALYTICS FOR SOCIAL SPAMMER DETECTION

In this chapter, I focus on the problem of exploiting content and relation information for social spammer detection. Due to the distinct characteristics of microblogging data, I focus on the social spammer detection in microblogging in this chapter. I will firstly review the background of this problem, and then formally define the problem and present the proposed method. The real-world dataset from Twitter will be used to evaluate the effectiveness of the proposed method by comparing with the state-of-the-art baselines.

3.1 Heterogeneous Information Sources in Social Media

Social media services are playing an important role in people's daily life. Because of the negative impact brought by social spammers, spammer detection has been studied in various online social networking (OSN) platforms [81, 15]. One effective method is to evaluate users' social reputation by social network analysis [13, 23]. The assumption behind the methods is that spammers cannot establish a large number of followers. Some social media systems, such as Twitter, feature unidirectional user binding, meaning anyone can follow anyone else without prior consent from the followee. Many users simply follow back when they are followed by someone for the sake of courtesy [102]. This phenomena in online social networks is called *reflexive reciprocity*, which makes it easier for spammers to imitate normal users in microblogging by quickly accumulating a large number of social relations. A recent study [34] on microblogging shows that spammers can successfully acquire a number of normal followers, especially those referred to as social capitalists who tend to increase their social capital by following back anyone who follows them.

Besides social network information, it is noted that microblogging provides additional content information, i.e., microblogging messages. Different from email spam detection, content analysis in microblogging for social spammer detection has been little studied due to the distinct characteristics of microblogging messages. First, microblogging messages are very short. For example, Twitter allows users to post messages up to 140 characters. Short messages cannot provide sufficient context information for effective similarity measure, the basis of many text processing methods [46]. Second, microblogging messages are very unstructured and noisy. In particular, when composing a message, users often prefer to use newly created abbreviations or acronyms. The slang words seldom appear in conventional text documents, but they do provide convenience for user-user communication. Although they provide a better user experience, unstructured expressions make it very difficult to accurately identify the semantic meanings of these messages. Last, microblogging messages are *networked* [110] in the sense that they are generated by users following some others in microblogging systems. The traditional assumption in many applications that data instances are independent and identically distributed (i.i.d.) is thus no longer valid for networked microblogging messages. The distinct features make traditional text analytics less applicable in microblogging platforms.

To address the new challenges posed by microblogging services, I propose to take advantage of both network and content information for social spammer detection in microblogging. In this chapter, I study the problem of social spammer detection in microblogging with network and content information. In essence, I investigate the following three questions:

- How do we model heterogeneous information sources, i.e., the network information and content information, properly in a unified framework?

- How do we seamlessly utilize both sources of information for the problem?

The solutions to these two challenges result in a new framework for Social Spammer Detection in Microblogging (*SSDM*). In particular, I employ a directed Laplacian formulation to model the refined social networks, and then integrate the network information into a sparse supervised formulation for the modeling of content information. Next, I will introduce the problem formulation.

3.2 Notations and Problem Formulation

In this section, I first introduce the notations used in the dissertation and then formally define the problem I study. Please note that, in the following sections, I will use the same notations to illustrate the proposed models.

Notation: The following notations are used in the dissertation. Matrices are denoted by boldface uppercase letters, vectors by boldface lowercase letters, and scalars by lower case letters. Let $||\mathbf{A}||$ denote the Euclidean norm, and $||\mathbf{A}||_F$ the Frobenius norm of the matrix $\mathbf{A}$. Specifically, $||\mathbf{A}||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{A}_{ij}^2}$. Let $\mathbf{A}^T$ denote the transpose of $\mathbf{A}$.

Let $\mathbf{U} = [\mathcal{G}, \mathbf{X}, \mathbf{Y}]$ be a target microblogging user set with social network information $\mathcal{G}$, content information of microblogging messages $\mathbf{X}$, and identity label matrix $\mathbf{Y}$. I use $\mathcal{G} = (V, E)$ to denote the social network, where nodes u and v in V represent microblogging users, and each directed edge $[u, v]$ in E represents a following relation from u to v. We do not have self links in the graph, i.e., $u \neq v$. I use $\mathbf{X} \in \mathbb{R}^{m \times n}$ to denote content information, i.e., messages posted by the users, where m is the number of textual features, and n is the number of users. I use $\mathbf{Y} \in \mathbb{R}^{n \times c}$ to denote the identity label matrix, where c is the number of identity labels. Following previous work on spammer detection [4, 63], I focus on classifying users as either spammers or normal users, i.e., $c = 2$. It is straightforward to extend this setting to a multi-class classification task.

With the defined notations, I formally define the problem of social spammer detection in microblogging as follows:

Given a set of microblogging users $\mathbf{U}$ *with social network information* $\mathcal{G}$*, content information* $\mathbf{X}$*, and identity label information* $\mathbf{Y}$ *of part of the users in the set (i.e., training data), I aim to learn a classifier* $\mathbf{W}$ *to automatically assign identity labels for unknown users (i.e., test data) as spammers or normal users.*

3.3 Exploiting Content and Relation Information for Social Spammer Detection

In this section, I first introduce how I model microblogging messages for each user and then discuss the modeling of social network information. Finally, I present a framework *SSDM* that considers both network and content information with its optimization algorithm.

3.3.1 Modeling Content Information

One widely used method for text analytics is Least Squares [61], which learns a linear model to fit the training data. The classification task can be performed by solving

$$\min_{\mathbf{W}} \ \frac{1}{2}||\mathbf{X}^T\mathbf{W} - \mathbf{Y}||_F^2, \tag{3.1}$$

where $\mathbf{X}$ is the content matrix of training data, $\mathbf{Y}$ is the label matrix, and $\mathbf{W} \in \mathbb{R}^{m\times 2}$ denotes the model I want to learn. This formulation is to minimize the learning error between the predicted value $\hat{\mathbf{Y}} = \mathbf{X}^T\mathbf{W}$ and the true value $\mathbf{Y}$ in the training data.

Microblogging messages are noisy and unstructured. The traditional text representation methods, like the "Bag of Words" or the N-gram model, often lead to the "curse of dimensionality." It is also observed that when people speed-read a text, they may not fully parse every word but instead seek a sparse representation with a few key phrases or words [73]. In addition, by providing some meaningful words rather than non-intuitive ones, it may help sociologists, security engineers, and even the public understand the motivation and behavior of social spammers. So I am motivated to exploit sparse learning [33], which allows better interpretability of the learning results for social spammer detection.

In many real-world applications, sparse learning methods have shown the effectiveness and been used to obtain a more efficient and interpretable model. One of the most widely used methods is the lasso [32], which introduces an ℓ_1-norm penalization on Least Squares. The classifier can be learned by solving the following optimization problem:

$$\min_{\mathbf{W}} \ \frac{1}{2}||\mathbf{X}^T\mathbf{W} - \mathbf{Y}||_F^2 + \lambda_1||\mathbf{W}||_1, \tag{3.2}$$

where $||\mathbf{W}||_1 = \sum_{i=1}^m \sum_{j=1}^c |\mathbf{W}_{ij}|$, and λ_1 is the sparse regularization parameter. The second term leads to a sparse representation of the learned model. As pointed out by Zou and Hastie [111], if there is a group of variables among which the pairwise correlations are very high, then the lasso tends to randomly select variables from this group. To make the sparse learning more stable, I further employ elastic net [111], which does automatic variable selection and continuous shrinkage, and can select groups of correlated variables. It is formulated by further adding a Frobenius norm regularization on the model as follows:

$$\min_{\mathbf{W}} \ \frac{1}{2}||\mathbf{X}^T\mathbf{W} - \mathbf{Y}||_F^2 + \lambda_1||\mathbf{W}||_1 + \frac{\lambda_2}{2}||\mathbf{W}||_F^2, \tag{3.3}$$

where λ_1 and λ_2 are positive parameters to control the sparsity and robustness of the model.

3.3.2 Modeling Social Network Information

To make use of network information, many methods assume that two nodes share a similar label when they are mutually connected in the network [19, 38, 110]. It has distinct features in microblogging. First, users have a directed following relation in microblogging. Second, spammers can easily follow a large number of normal microblogging users within a short time. Thus the existing methods are not suitable to this problem.

I first refine the social relations in the social network. Given the social network information $\mathcal{G}$ and the identity label matrix $\mathbf{Y}$, there are four kinds of following relations:

[spammer, spammer], [normal, normal], [normal, spammer], and [spammer, normal].

Since the fourth relation can be easily faked by spammers, I make use of the first three relations in the proposed framework. Now I introduce how to represent and model the social network information in detail.

The adjacency matrix $\mathbf{G} \in \mathbb{R}^{n \times n}$ is used to represent the refined social network $\mathcal{G}$, and it is defined as

$$\mathbf{G}(u, v) = \begin{cases} 1 & \text{if [u, v] is among the first three relations} \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

where u and v are nodes, and $[u, v]$ is a directed edge in the graph $\mathcal{G}$. The in-degree of the node u is defined as $\mathbf{d}_u^{in} = \sum_{[v,u]} \mathbf{G}(v, u)$, and the out-degree of the node u is defined as $\mathbf{d}_u^{out} = \sum_{[u,v]} \mathbf{G}(u, v)$. Let $\mathbf{P}$ be the transition probability matrix of random walk in a given graph with $\mathbf{P}(u, v) = \mathbf{G}(u, v)/\mathbf{d}_u^{out}$ [109]. The random walk has a stationary distribution $\boldsymbol{\pi}$, which satisfy $\sum_{u \in V} \boldsymbol{\pi}(u) = 1$, $\boldsymbol{\pi}(v) = \sum_{[u,v]} \boldsymbol{\pi}(u)\mathbf{P}(u, v)$ [20], and $\boldsymbol{\pi}(u) > 0$ for all $u \in V$.

The key idea here is that I employ network information to smooth the learned model. It can be mathematically formulated as minimizing

$$\mathcal{R}_S = \frac{1}{2} \sum_{[u,v] \in E} \boldsymbol{\pi}(u)\mathbf{P}(u, v) ||\hat{\mathbf{Y}}_u - \hat{\mathbf{Y}}_v||^2, \tag{3.5}$$

where $\hat{\mathbf{Y}}_u$ denotes the predicted label of user u, and $\hat{\mathbf{Y}}_v$ the predicted label of user v. The loss function will incur a penalty if two users have different predicted labels when they are close to each other in the graph.

Let $\mathbf{\Pi}$ denote a diagonal matrix with $\mathbf{\Pi}(u, u) = \boldsymbol{\pi}(u)$.

Theorem 1 *The formulation in Eq. (3.5) is equivalent to the following objective function:*

$$\mathcal{R}_S = tr(\hat{\mathbf{Y}} \mathcal{L} \hat{\mathbf{Y}}^T), \tag{3.6}$$

where the Laplacian matrix [20] $\mathcal{L}$ is defined as

$$\mathcal{L} = \mathbf{\Pi} - \frac{\mathbf{\Pi P} + \mathbf{P}^T \mathbf{\Pi}}{2}. \tag{3.7}$$

Proof. The proof is straightforward and can be also found in previous work [20, 109]. □

3.3.3 Social Spammer Detection in Microblogging

Many existing text classification methods assume that instances are independent and identically distributed (i.i.d.). They focus on either building a sophisticated feature space or employing effective classifiers to achieve better classification performance, without taking advantage of the fact that the instances are networked with each other. In the problem of social spammer detection, microblogging users are connected via social networks. I propose to consider both network and content information in a unified model.

Since $\hat{\mathbf{Y}} = \mathbf{X}^T\mathbf{W}$, Eq. (3.6) can be easily rewritten as

$$\mathcal{R}_S = tr(\mathbf{W}^T\mathbf{X}\mathcal{L}\mathbf{X}^T\mathbf{W}). \tag{3.8}$$

By considering both network and content information, the social spammer detection can be formulated as the following optimization problem:

$$\min_{\mathbf{W}} \ \frac{1}{2}||\mathbf{X}^T\mathbf{W} - \mathbf{Y}||_F^2 + \lambda_1||\mathbf{W}||_1 + \frac{\lambda_2}{2}||\mathbf{W}||_F^2 + \frac{\lambda_s}{2}tr(\mathbf{W}^T\mathbf{X}\mathcal{L}\mathbf{X}^T\mathbf{W}). \tag{3.9}$$

By solving Eq. (3.9), the identity label of each unknown target user $\mathbf{x}$ can be predicted by

$$\arg\max_{i\in\{spammer,normal\}} \quad \mathbf{x}^T\mathbf{w}_i. \tag{3.10}$$

Next I introduce an efficient algorithm to solve the optimization problem in Eq. (3.9).

The optimization problem in Eq. (3.9) is convex and non-smooth. Following [68, 76], the basic idea of the proposed algorithm is to reformulate the non-smooth optimization problem as an equivalent smooth convex optimization problem.

Lemma 1 $||\mathbf{W}||_1$ *is a valid norm.*

Proof. It is easy to verify that $||\mathbf{W}||_1$ satisfies the three conditions of a valid norm, including the triangle inequality $||\mathbf{A}||_1 + ||\mathbf{B}||_1 \leq ||\mathbf{A} + \mathbf{B}||_1$, which completes the proof. □

Theorem 2 *Eq. (3.9) can be reformulated as a constrained smooth convex optimization problem:*

$$\min_{\mathbf{W}\in\mathcal{Z}} \mathcal{O}(\mathbf{W}) = \frac{1}{2}\|\mathbf{X}^T\mathbf{W} - \mathbf{Y}\|_F^2 + \frac{\lambda_2}{2}\|\mathbf{W}\|_F^2 + \frac{\lambda_s}{2} tr(\mathbf{W}^T\mathbf{X}\mathcal{L}\mathbf{X}^T\mathbf{W}), \tag{3.11}$$

where

$$\mathcal{Z} = \{\mathbf{W}| \ \|\mathbf{W}\|_1 \leq z\}, \tag{3.12}$$

and $z \geq 0$ is the radius of the ℓ_1-ball. Note that λ_1 and z have a one-to-one correspondence between each other.

Proof. Since $\|\mathbf{W}\|_1$ is a valid norm, it defines a closed and convex set $\mathcal{Z}$. The Hessian matrix of the reformulated objective function $\mathcal{O}(\mathbf{W})$ is positive semi-definite. Thus the optimization problem in Eq. (3.11) is convex and differentiable. This problem defines a convex and differentiable function $\mathcal{O}(\mathbf{W})$ in a closed and convex set $\mathcal{Z}$. Thus the reformulated function is a constrained smooth convex optimization problem, which completes the proof. □

A widely used method, proximal gradient descent [56], is employed to optimize the above constrained smooth convex problem. The method solves the problem by updating the following,

$$\mathbf{W}_{t+1} = \arg\min_{\mathbf{W}\in\mathcal{Z}} M_{\gamma,\mathbf{W}_t}(\mathbf{W}), \tag{3.13}$$

where $M_{\gamma,\mathbf{W}_t}(\mathbf{W})$ is the Euclidean projection [12, Chapter 8.1], which is defined as

$$M_{\gamma,\mathbf{W}_t}(\mathbf{W}) = \mathcal{O}(\mathbf{W}_t) + \langle\nabla\mathcal{O}(\mathbf{W}_t), \mathbf{W} - \mathbf{W}_t\rangle + \frac{\gamma}{2}\|\mathbf{W} - \mathbf{W}_t\|_F^2, \tag{3.14}$$

where γ is the step size, and

$$\nabla\mathcal{O}(\mathbf{W}_t) = \mathbf{X}\mathbf{X}^T\mathbf{W}_t - \mathbf{X}\mathbf{Y} + \lambda_2\mathbf{W} + \lambda_s\mathbf{X}\mathcal{L}\mathbf{X}^T\mathbf{W}_t. \tag{3.15}$$

Let $\mathbf{U}_t = \mathbf{W}_t - \frac{1}{\gamma}\nabla O(\mathbf{W}_t)$. The Euclidean projection has a closed-form solution [68] as follows:

$$\mathbf{w}_{t+1}^j = \begin{cases} (1 - \frac{\lambda_1}{\gamma\|\mathbf{u}_t^j\|})\mathbf{u}_t^j & \text{if } \|\mathbf{u}_t^j\| \geq \frac{\lambda_1}{\gamma} \\ 0 & \text{otherwise} \end{cases} \tag{3.16}$$

where $\mathbf{u}_t^j$, $\mathbf{w}^j$ and $\mathbf{w}_t^j$ are the j-th rows of $\mathbf{U}_t$, $\mathbf{W}$ and $\mathbf{W}_t$, respectively.

Based on this algorithm discussed above, I can have an efficient and optimal solution to the convex optimization problem. Similar to the proof in [68], it is easy to verify that the convergence rate of the proposed algorithm is $O(\frac{1}{\sqrt{\epsilon}})$ for achieving an accuracy of ϵ.

3.4 Experiments

In this section, I conduct experiments to evaluate the effectiveness of the proposed framework *SSDM*. Through the experiments, I aim to answer the following two questions:

1. How effective is the proposed framework compared with other methods of social spammer detection?

2. What are the effects of the social network and content information on the social spammer detection?

I begin by introducing the dataset and experimental setup and then compare the performance of different spammer detection methods. Finally, I study the effects of important parameters on the proposed method.

3.4.1 Experimental Settings

I first introduce the real-world Twitter dataset used in the experiment. A data crawling process, which is similar to [94, 105, 110], is employed to construct the dataset. I first crawled a Twitter dataset from July 2012 to September 2012 via the Twitter Search API.[1]

[1] https://dev.twitter.com/docs/api/1/get/search/

Table 3.1: Summary of the Experimental Dataset

# Spammers	# Normal Users	Max Degree of Users
2,118	10,335	1,025
# Tweets	# Unigrams	Min Degree of Users
380,799	21,388	3

The users that were suspended by Twitter during this period are considered as the gold standard [94] of spammers in the experiment. I then randomly sampled the normal users which have social relations with the spammers. According to the literature of spammer detection, the two classes are imbalanced, i.e., the number of normal users I sampled is much greater than that of spammers in the dataset. I finally remove stop-words and perform stemming for all the tweets. The statistics of the dataset is presented in Table 3.1.

I follow standard experiment settings used in [4, 110] to evaluate the performance of spammer detection methods. In particular, I apply different methods on the Twitter dataset. Precision, recall, and F_1-measure are used as the performance metrics.

There are three positive parameters involved in the experiments, including λ_1, λ_2, and λ_s in Eq. (3.9). λ_1 is to control the sparsity of the learned model, λ_2 is the parameter to make the learned model more robust, and λ_s is to control the contribution of network information. As a common practice, all the parameters can be tuned via cross-validation with validation data. In the experiments, I empirically set $\lambda_1 = 0.1$, $\lambda_2 = 0.1$, and $\lambda_s = 0.1$ for general experiment purposes. The effects of the parameters on the learning model will be further discussed in Section 3.4.4.

3.4.2 Effectiveness of the Proposed Model

This set of experiments is to answer the first question asked in the beginning of Section 3.4. I compare the proposed method *SSDM* with the following baseline methods. All the methods utilize both content and network information in different ways.

- *LS_Content_SN*: the Least Squares [61] is a widely used classifier for i.i.d. data. I combine the content matrix **X** and adjacency matrix **G** of the social network together for user representation.

- *EN_Content_SN*: the elastic net is one of the most effective sparse learning methods [111], and it is applied on the same data matrix as the first baseline.

- *SMF_UniSN*: a multi-label informed latent semantic indexing [107, 110] is used to model the content information, and undirected graph Laplacian [19] is used to incorporate the network information. This is the state-of-the-art method for spammer detection in an undirected social network. In the experiment, I convert the directed graph to an undirected one with $\mathbf{G} = max(\mathbf{G}, \mathbf{G}^T)$.

- *SSDM*: the proposed method for spammer detection.

I present the experimental results of the methods in Table 3.2. In the experiment, I use five-fold cross validation for all the methods. To avoid effects brought by the size of the training data, I conduct two sets of experiments with different numbers of training samples. In each round of the experiment, 80% of the whole dataset is held for training. In the table, "50% of Training Data" means that I randomly chose 50% of the 80%, thus using 40% of the whole dataset for training. Also, "gain" represents the percentage improvement of the methods in comparison with the first baseline method *LS_Content_SN*. In the experiment, each result denotes an average of 10 test runs. By comparing the results of different methods, I draw the following observations:

Table 3.2: Social Spammer Detection Results

	50% of the Training Data			*100% of the Training Data*		
	Precision	Recall	F_1-measure (gain)	Precision	Recall	F_1-measure (gain)
LS_Content_SN	0.786	0.843	0.813 (N.A.)	0.793	0.850	0.821 (N.A.)
EN_Content_SN	0.801	0.872	0.835 (+2.69%)	0.836	0.891	0.863 (+5.09%)
SMF_UniSN	0.804	0.889	0.845 (+3.87%)	0.844	0.915	0.878 (+6.92%)
SSDM	0.852	0.896	0.873 (+7.40%)	0.865	0.939	0.901 (+9.73%)

(1) From Table 3.2, we can observe that the proposed framework *SSDM* consistently outperforms other baseline methods using all metrics with different sizes of training data. The proposed method achieves better performance than the state-of-the-art method *SMF_UniSN*. I apply two-sample one-tail t-tests to compare *SSDM* with the three baseline methods. The experiment results demonstrate that *SSDM* performs statistically significantly better (with the significance level $\alpha = 0.01$) than the three methods. This indicates that, compared with other methods, the proposed model successfully utilizes both content and network information for social spammer detection.

(2) Among the three baseline methods, *LS_Content_SN* achieves the worst performance. With the introduction of sparsity regularization, *EN_Content_SN* has performance improvement. This demonstrates that sparse learning is effective to handle the noisy and high-dimensional data in microblogging. *SMF_UniSN* achieves the best performance.

(3) We also compare the AUC value of the baseline methods and the proposed *SSDM*. Among all of the four methods, the proposed method *SSDM* achieves the highest AUC (0.913), which means that the proposed method is not sensitive with different discrimination threshold.

From the results above, the methods perform differently in social spammer detection. In most cases, the simple combination of content and network information does not work well. It suggests that the way of using the two kinds of information is important. The superior performance of the proposed method answers the first question that, compared with other methods, *SSDM* is effective in social spammer detection.

3.4.3 Effectiveness of an Individual Information Source

This subsection is to study the importance of each kind of information and accordingly answer the second question asked in the beginning of this section. I compare the proposed method with the following two groups of four methods:

- *Content-based methods*: support vector machine (SVM) and elastic net (EN) are employed for spammer detection based on content information only.
- *Network-based methods*: SVM and EN are employed based on network information, which is represented as the adjacency matrix of the social network.

The performance of the proposed framework *SSDM* is compared with the methods with only one type of information on the Twitter dataset. The results are plotted in Figure 3.1. The first four bars represent the performance of the two representative methods *SVM* and *EN* with one type of information, respectively. The last is the proposed method *SSDM*.

From the figure, it shows that, with the integration of content and network information in a unified model, the proposed framework *SSDM* achieves better performance than those with only one kind of information. Among the four baseline methods, *SVM_Content* and *EN_Content* have comparable performance. They significantly outperform the other two methods *SVM_SN* and *EN_SN*. This demonstrates that, in this experiment, content information is more effective than social network information. I need a more sophisticated way

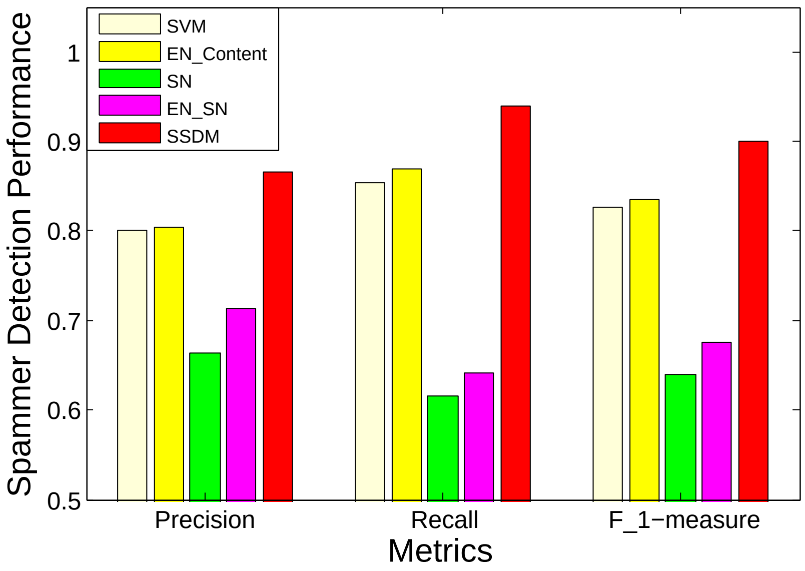


Figure 3.1: Social Spammer Detection Performance

to represent social network information for social spammer detection. Simply employing neighbors of a user for representation does not work well.

The results show that the methods based on network information do not have good performance in social spammer detection. It suggests that the way of integrating social network information is important. The superior performance of the proposed method *SSDM* further validates its excellent use of both network and content information in a unified way.

3.4.4 Discussion

There two important parameters, i.e., λ_1 and λ_s, involved in the proposed formulation and need to be further explored. λ_1 is to control the sparseness of the learned model, and λ_s is to control the contribution of social network information to the model. I now conduct experiments to compare the social spammer detection performance of the proposed *SSDM* on the Twitter dataset with different parameter settings. The social spammer detection results (F_1-measure) of *SSDM* with different parameter settings on the dataset are plotted

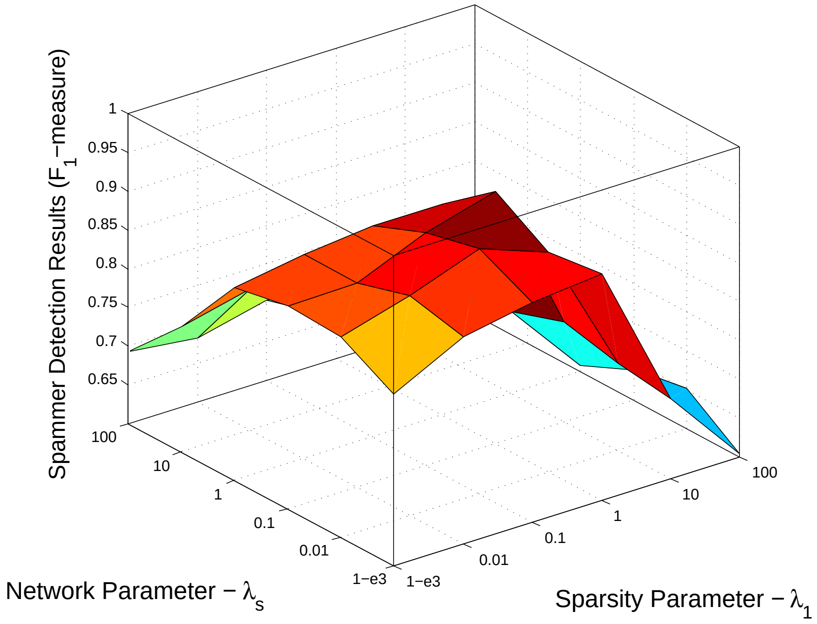


Figure 3.2: Impact of the Sparsity Parameter (λ_1) and the Network Parameter (λ_s)

in Figure 3.2. In the figure, performance of *SSDM* improves as the parameters λ_1 and λ_s increase, and reaches a peak at $\lambda_1 = 0.1$ and $\lambda_s = 1$. When $\lambda_1 > 0.1$ or $\lambda_s > 1$, the performance of *SSDM* declines. Generally, the performance is not very sensitive to λ_1 when it is in a reasonable range [0.01, 10]. The performance changes significantly when $\lambda_s > 1$. The results suggest that the proposed framework can achieve relatively good performance when the parameters are in the range [0.01, 1].

3.5 Summary

In this chapter, I investigate how to seamlessly integrate the content and network information to perform effective social spammer detection. In particular, the proposed framework models both types of information in a unified way. Experiments on a real Twitter dataset show that the proposed *SSDM* framework can effectively integrate both kinds of information to outperform the state-of-the-art methods.

Chapter 4

CONTEXTUAL DATA ANALYTICS FOR SOCIAL SPAMMER DETECTION

In this chapter, I focus on the problem of exploiting contextual sentiment information for social spammer detection. I will firstly review the background of this problem that why sentiment information could be potentially useful for social spammer detection. And then I formally define the problem and introduce the proposed method. Real-world datasets from Twitter are used to evaluate the effectiveness of the proposed method by comparing with the state-of-the-art baseline methods.

4.1 Contextual Information in Social Media

While social spammer detection is a relatively novel problem, understanding and detecting deception has been extensively studied in psychology and social sciences. It is well-established that *microexpressions* [40] play a distinct role in detecting deception. Microexpression is an involuntary facial expression of humans according to sentiments experienced. It usually occurs when a person is consciously trying to conceal all signs of how he or she is feeling [40]. Ekman [29] reported that facial and emotional "microexpressions" could be useful to assist in lie detection after testing a total of 20,000 people [30] from all walks of life. Also, as pointed out by Matsumoto *et al.* [74], one may not conclude that someone is lying if a microexpression is detected but that there is more to the story than is being told. Inspired by the psychological findings, I explore whether the utilization of sentiment information could help capture deceptions of the social spammers.

In this chapter, I focus on the problem of utilizing sentiment information for effective social spammer detection. Specifically, I am particularly interested in answering the following questions:

- Is sentiment information potentially useful for social spammer detection?
- How can sentiment information be explicitly represented and incorporated for social spammer detection?
- Is the integration of sentiment analysis helpful for the studied problem?

To answer these questions, it results in a novel framework for social Spammer Detection with Sentiment information (*SDS*). In particular, I first investigate whether sentiment differences between spammers and normal users exist in social media data. Then I discuss how to model sentiment information, combined with content and network information, in a novel social spammer detection framework. Finally, I conduct extensive experiments to evaluate the proposed model.

4.2 Problem Statement

In this section, I formally define the problem of utilizing sentiment information for social spammer detection.

One distinct feature of social media data is that it provides abundant contextual information other than social networks. The problem I studied is different from traditional spammer detection in social networks since the latter typically only considers either the content or network information [13, 48]. In this section, I first formally define the problem of social spammer detection with sentiment information.

Let $\mathbf{S} = [\mathbf{X}, \mathcal{G}, \mathbf{Y}]$ be a target user set with content information $\mathbf{X}$, social network information $\mathcal{G}$ and identity label matrix $\mathbf{Y}$. I use user-word matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ to denote content information, i.e., posts written by the users, where n is the number of users, and m is the number of textual features. I use $\mathcal{G} = (V, E)$ to denote the social network, where nodes $v \in V$ represent social media users, and each directed edge between two nodes $[u, v] \in E$ represents a following relation from u to v. There are no self-links in the graph, i.e., $u \neq v$.

$\mathbf{Y} \in \mathbb{R}^{n \times c}$ denotes the identity label matrix, where c is the number of possible identity labels. In this chapter, I focus on the binary classification problem, i.e., $c = 2$ and the users will be classified as spammers or normal users. It is practical to extend this setting to a multi-class classification task. Given another corpus of posts $\mathbf{C} \in \mathbb{R}^{t \times m}$ with sentiment labels, where t is the number of posts, and m is the number of textual features. I use $\mathbf{s} \in [-1, 1]^t$ to represent the sentiment polarity labels of the corresponding social media posts. For example, $\mathbf{s}(i) = 1$ represents that sentiment of the i^{th} post in the corpus is positive, and $\mathbf{s}(i) = -1$ negative.

I now formally define the problem as follows: *Given a set of social media users* $\mathbf{S}$ *with content information* $\mathbf{X}$*, social network information* $\mathcal{G}$*, and identity label information* $\mathbf{Y}$ *of part of the users in the set (i.e., training data), I can also learn the sentiment information from another set of labeled posts* $[\mathbf{C}, \mathbf{s}]$*, the goal is to learn a model to automatically assign identity labels for unknown users (i.e., test data) as spammers or normal users.*

4.3 Sentiment Correlation Analysis

As discussed above, the major motivation of this study is to investigate if sentiment information is useful for social spammer detection. Before proceeding further, I first introduce real-world datasets used in this work and examine whether sentiment information has any potential impact for social spammer detection.

Three Twitter datasets are used in this study. The first two contain labels for social spammer detection, i.e., TAMU Social Honeypots and Twitter Suspended Spammers, and the third one Stanford Twitter Sentiment has sentiment labels. Now I introduce the three datasets in detail.

TAMU Social Honeypots Dataset (TUSH):[1] Lee *et al.* [64] created a collection of 41,499 Twitter users with identity labels as spammers and normal users. The dataset was collected from December 30, 2009 to August 2, 2010 on Twitter. It consists of users,

[1] `http://infolab.tamu.edu/data/`

Table 4.1: Statistics of the Datasets

Statistics	*TUSH*	*TSS*
# of Spammers	16,841	4,005
# of Normal Users	13,697	15,832
# of Unigrams	31,004	18,055

their number of followers and posted tweets. I further refined the dataset according to users' social relation information, which is a complete follower graph[2] crawled by Kwak *et al.* [60] during July 2009. According to the social network, I filter the users who post less than two tweets or have less than two friends in the dataset. Finally, it leaves a corpus of 30,538 users that consists of 16,841 spammers and 13,697 normal users. This dataset has balanced number of spammers and normal users.

Twitter Suspended Spammers Dataset (TSS): I used a data construction process, which is similar to [105, 110], to build this dataset. I first crawled a Twitter dataset from August 5, 2013 to October 11, 2013 using the Twitter Search API.[3] I examined all of the crawled users at the end of the crawling process. The users that were suspended by Twitter during this period are considered as the gold standard [110] of spammers in the experiment. I then randomly sampled normal users which have social relations with the spammers. To consider effects brought by different class distribution, according to the literature of social spammer detection [63], I made the two classes in TSS imbalanced, i.e., the number of normal users I sampled is much greater than that of spammers in the dataset. In addition, users that post less than two tweets or have less than two friends in the whole dataset are removed. Finally, it leaves a corpus of 19,837 users that consists of 4,005 spammers and

[2] http://an.kaist.ac.kr/traces/WWW2010.html/

[3] http://dev.twitter.com/docs/api/

15,832 normal users. A standard procedure is used for data preprocessing on both datasets. All of the non-English tweets are filtered out from the datasets. I remove stop-words and perform stemming for all the tweets. The unigram model is employed to construct the feature space, tf-idf is used as the feature weight. The statistics of the datasets are presented in Table 4.1.

Stanford Twitter Sentiment (SENT)[4]: Go *et al.* [35] created a collection of 40,216 tweets with polarity sentiment labels to train a sentiment classifier. The tweets in the dataset are crawled between April 6, 2009 and June 25, 2009. All the tweets and corresponding sentiment labels in the dataset are used to learn a model for sentiment analysis.

A standard method is used to compute the sentiment score of each user. In particular, a supervised sentiment analysis model is learned based on the labeled dataset SENT, and I then apply the learned model to compute the sentiment score of users in the two datasets TUSH and TSS. Pang and Lee [84] conducted experiments to study the effectiveness of different methods on sentiment analysis. It shows that machine learning techniques can achieve good performance on benchmark datasets. Following widely used sentiment analysis methods introduced in [84, 67, 51], a linear regression [32] is employed to fit the learned model to sentiment labels $\mathbf{s}$. The linear regression aims to learn a model by solving the following optimization problem:

$$\min_{\mathbf{w}} \quad ||\mathbf{C}\mathbf{w} - \mathbf{s}||^2, \tag{4.1}$$

where $\mathbf{C}$ represents the content matrix of SENT dataset, $\mathbf{w}$ represents the learned coefficients of the features, and $\mathbf{s}$ denotes the sentiment labels of the posts in $\mathbf{C}$. This formulation is a traditional supervised learning method, and it has a closed-form solution: $\mathbf{w} = (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{s}$. By solving Eq. (4.1), the sentiment score of a user u can be computed by $\mathbf{X}(u)\mathbf{w}$.

[4] http://www.stanford.edu/~alecmgo/cs224n/

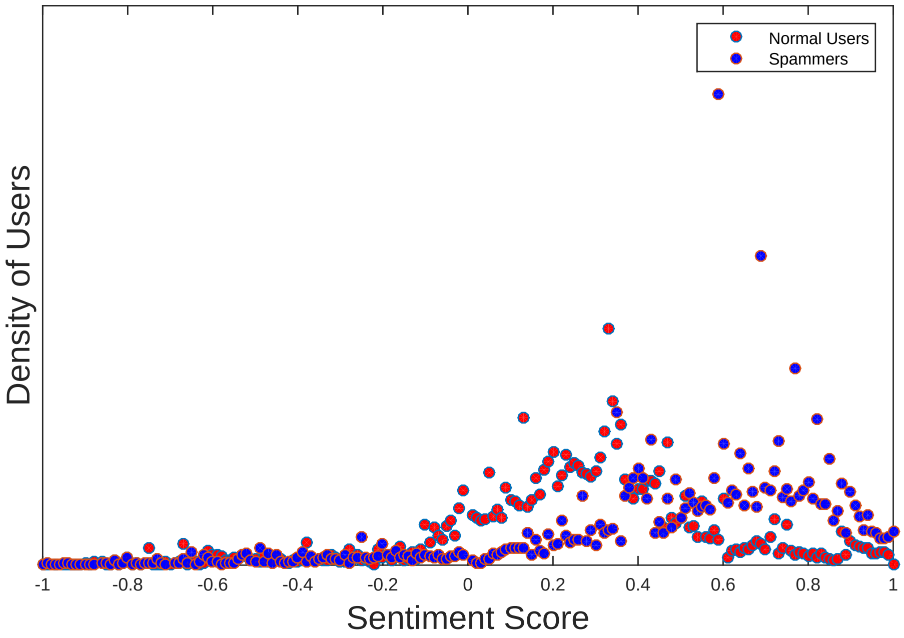


Figure 4.1: Sentiment Score Distribution

The sentiment score of each user in the two datasets are calculated. The sentiment scores are normalized in the range of $[-1, 1]$. I plot the polarity score distributions of spammers and normal users on the TUSH dataset in Figure 4.1. In the figure, x axis represents the sentiment score and y axis the density of users who have the exact sentiment score. Red dots denotes the sentiment score distribution of normal users and blue dots the distribution of spammers. From the figure, I can observe two normal-like distributions for spammers and normal users. The two distributions center with different mean values and show clearly different patterns. It suggests that the sentiment patterns of normal users and spammers are different. Similar results have been observed on the TSS dataset.

Table 4.2: T-Test Results (P-Values) to Verify Microexpressions

	TUSH	TSS
Microexpressions	<0.938e-9	<1.011e-15

4.3.1 Evaluating Usefulness of Sentiment Information

The preliminary results in Section 4.3 show that the sentiment distributions of spammers and normal users are different. I now further verify whether this observation is potential useful for the studied problem.

The psychological finding of microexpression suggests that sentiments of spammers are different from normal users. The assumption is that the sentiments of two users with the same identity, i.e., both are spammers or normal users, have higher probability to be consistent than those of two random users. I use hypothesis testing to validate whether this assumption of sentiment consistency holds in the two Twitter datasets. I first define the sentiment difference score $d(i, j)$ between two users as

$$d(i, j) = \|\mathbf{s}(i) - \mathbf{s}(j)\|_2, \tag{4.2}$$

where $\mathbf{s}(i)$ and $\mathbf{s}(j)$ represent sentiment scores of the two users. The sentiment scores are computed by the method I introduced in Section 4.3.

Then, two vectors $\mathbf{s}_c$ and $\mathbf{s}_r$ with an equal number of elements are constructed. Each element of the first vector $\mathbf{s}_c$ is calculated by Eq. (4.2), where $\mathbf{s}(i)$ and $\mathbf{s}(j)$ are users with the same identity. Each element of the second vector represents the sentient difference score between $\mathbf{s}(i)$ and $\mathbf{s}(r)$, which denotes the sentiment score of another randomly selected user. I form a two-sample one-tail t-test to validate the assumption. I test whether there is sufficient evidence to support the hypothesis that sentiment difference of the first group is greater or equal than that of the second. The null hypothesis and alternative hypothesis are

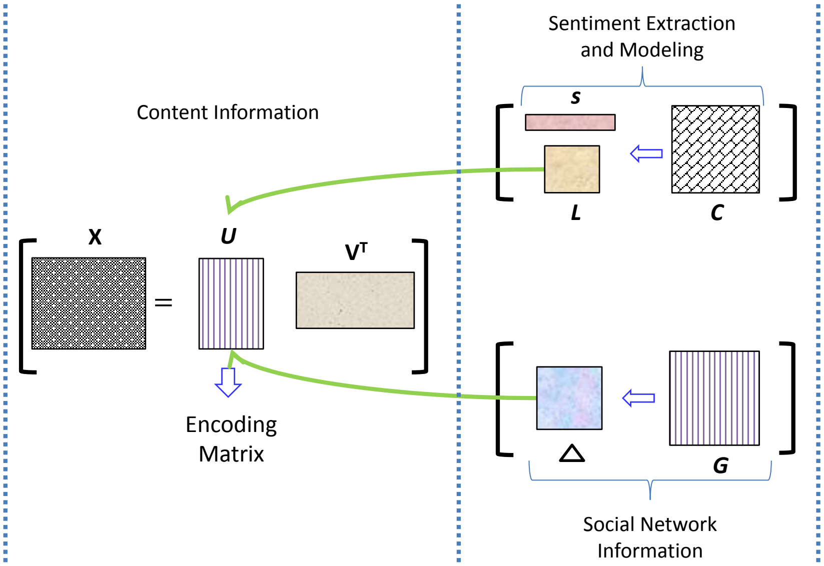


Figure 4.2: Illustration of the Spammer Detection Framework

formulated as follows:

$$\begin{aligned} H_0 &: \mu_c - \mu_r \geq 0 \\ H_1 &: \mu_c - \mu_r < 0 \end{aligned} \tag{4.3}$$

where μ_c and μ_r represent the sample means of sentiment difference scores in the two groups, respectively.

The hypothesis testing results, p-values, are summarized in Table 4.2. The results suggest that there is strong statistical evidence, with significance level $\alpha = 0.01$, to reject the null hypothesis on the two datasets. In other words, I validate the assumption in the two datasets. This exploratory study paves the way for the next step: how to explicitly model and utilize the sentiment information for social spammer detection.

4.4 Exploiting Sentiment Information for Social Spammer Detection

In this section, I introduce the proposed framework that incorporates sentiment, content and social network information for social spammer detection in detail.

The work flow of the proposed framework is plotted in Figure 4.2. From the figure, I can see that the whole framework consists of three components. The left part represents modeling of content information. There are two constraints on the learned factor matrix **U** which is derived from content information. As shown in the upper right part of the figure, the first constraint is from sentiment information $\mathcal{L}$, which is learned from an independent sentiment related source **C**. As shown in the lower right part of the figure, the second constraint is learned from social network information **G**. In this section, I first discuss how to model content information, and then introduce the modeling of sentiment and network information to detect social spammers. Finally, I present the framework that considers the three types of information as well as its computational algorithm for social spammer detection.

4.4.1 Matrix Factorization for Content Modeling

Unlike spam detection in platforms such as email and SMS, although content analysis is abundant in social media, it has been little studied for spammer detection. To make use of content information, a straightforward way is to learn a supervised model based on labeled data, and apply the learned model for spammer detection. However, this method yields two problems due to the unstructured and noisy content information in social media. First, text representation models, like n-gram model, often lead to a high-dimensional feature space because of the large size of data and vocabulary. Second, in addition to the short form of texts, abbreviations and acronyms are widely used in social media, thus making the data representation very sparse [46]. These distinct characteristics of social media data make traditional text analytics less applicable for the task.

To tackle the problems, I propose to model the content information from topic-level instead of learning word-level knowledge. Motivated by previous work on topic modeling [8], a user's posts usually focus on a few topics, resulting in **X** very sparse and low-rank.

The proposed method is built on a non-negative matrix factorization model (NMF) [62]. NMF is to seek a more compact but accurate low-rank representation of the users by solving the following optimization problem:

$$\min_{\mathbf{U},\mathbf{V}\geq 0} \quad \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2, \tag{4.4}$$

where $\mathbf{X}$ is the content matrix, $\mathbf{U} \in \mathbb{R}^{n\times r}$ with $r \ll m$ is an encoding matrix that indicates a low-rank user representation in a topic space and $\mathbf{V} \in \mathbb{R}^{m\times r}$ is a mixing matrix. Both $\mathbf{U}$ and $\mathbf{V}$ are non-negative factor matrices to be learned.

There are several nice properties by using matrix factorization [39, 90] based methods for content modeling : (1) this model has a nice probabilistic interpretation with Gaussian noise; (2) many existing optimization methods can be used to provide a well-worked optimal solution; (3) it can be scaled to a large number of users, which is a common setting in social media; (4) this formulation is flexible and allows us to introduce prior knowledge such as sentiment information and social network information.

4.4.2 Sentiment Information Modeling

The observation introduced in Section 4.3 suggests that the sentiments of two users with the same identity label have higher probability to be consistent. Based on this observation, I propose to model the sentiment information with graph Laplacian [19]. I construct an undirected graph $\mathbf{G}_S$ based on sentiment information of the users. In the graph, each node represents a user and each edge represents the sentiment correlation between two users. The adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ of the constructed graph $\mathbf{G}_S$ is formulated as the following:

$$\mathbf{A}(i,j) = \begin{cases} 1 & \text{if } u_i \in \mathcal{N}(u_j)\ or\ u_j \in \mathcal{N}(u_i) \\ 0 & \text{otherwise .} \end{cases} \tag{4.5}$$

where u_i and u_j are nodes, and $\mathcal{N}(u_i)$ represents the k-nearest neighbor of the user u_i in terms of sentiment information. As I discussed in Section 4.3, a model $\mathbf{w}$ can be learned by minimizing the objective function in Eq. (4.1), and sentiment score of a user u can be computed as $\mathbf{X}(u)\mathbf{w}$. It is noted that this study is not confined to any specific sentiment analysis tools. It is practical to employ other sentiment analysis methods, e.g., lexicon-based method [69], to compute the sentiment score of each user. Since I aim to model the mutual sentiment correlation between two users, the adjacency matrix in the formulation is symmetric.

The key idea of utilizing graph Laplacian to model the sentiment information is that if two nodes are close in the graph, i.e., their sentiment scores are close to each other, the representations of the two users should be similar. It can be formulated as minimizing the following loss function:

$$\mathcal{R}_S = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\mathbf{U}_i - \mathbf{U}_j)^2\mathbf{A}(i,j)\ , \tag{4.6}$$

where n is the number of users in the graph, $\mathbf{U}_i$ denotes representation of the i^{th} user, and $\mathbf{U}_j$ the j^{th} user. This loss function will incur a penalty if two users have different representations when they are close to each other in the constructed graph. Let $\mathbf{D} \in \mathbb{R}^{n\times n}$ denote a diagonal matrix, and its diagonal element is the degree of a user in the adjacency matrix $\mathbf{A}$, i.e., $\mathbf{D}(i,i) = \sum_{j=1}^{n}\mathbf{A}(i,j)$. It is easy to verify that the formulation in Eq. (4.6) can be rewritten as:

$$\begin{aligned}\mathcal{R}_S &= \sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{U}_i\mathbf{A}(i,j)\mathbf{U}_i^T - \sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{U}_i\mathbf{A}(i,j)\mathbf{U}_j^T \\ &= \sum_{i=1}^{n}\mathbf{U}_i\mathbf{D}(i,i)\mathbf{U}_i^T - \sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{U}_i\mathbf{A}(i,j)\mathbf{U}_j^T \\ &= Tr(\mathbf{U}^T(\mathbf{D}-\mathbf{A})\mathbf{U}) \\ &= Tr(\mathbf{U}^T\mathcal{L}\mathbf{U}). \end{aligned} \tag{4.7}$$

Besides sentiment information, abundant social network information is available in social media for social spammer detection. Next, I introduce how to model the social network information for the studied problem.

4.4.3 Social Network Information Modeling

Many efforts have been devoted to model social network information in various applications such as recommender systems [39] and trust prediction [90]. Existing methods often assume that representations of two nodes are close when they are connected with each other in the network [110, 19]. This assumption does not hold in many social media services. For example, some social media services such as microblogging allow directed following relations between users without mutual consent. In addition, as I discussed, spammers can easily follow a large number of normal users within a short time. The characteristics of the social media data make existing methods not suitable to this task.

I propose to use a variant of directed graph Laplacian to model network information. Given the social network information $\mathcal{G}$ and the identity labels $\mathbf{Y}$, four kinds of following relations can be extracted: [spammer, spammer], [normal, spammer], [normal, normal], and [spammer, normal]. Since the fourth relation [spammer, normal] can be easily faked by spammers, I only make use of the first three relations in the proposed framework. Note that this is a general setting in different social networks. In undirected social networks, e.g., Facebook, it is easy to convert the undirected graph into a direct setting. Now I introduce how to represent and model the social network information. The adjacency matrix $\mathbf{G} \in \mathbb{R}^{n \times n}$ is used to represent the refined directed social network $\mathcal{G}$, and it is defined as

$$\mathbf{G}(i, j) = \begin{cases} 1 & \text{if } [u_i, u_j] \text{ is among the first three relations} \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

where u_i and u_j represent the i^{th} and j^{th} users, and $[u_i, u_j]$ is a directed edge in the graph $\mathcal{G}$.

In-degree of the node u_i the social network is defined as $\mathbf{d}_i^{in} = \sum_{[u_j, u_i]} \mathbf{G}(j, i)$, and out-degree of the node u is defined as $\mathbf{d}_i^{out} = \sum_{[u_i, u_j]} \mathbf{G}(i, j)$. Let $\mathbf{P}$ be the transition probability matrix of random walk in a given graph with $\mathbf{P}(i, j) = \mathbf{G}(i, j)/\mathbf{d}_i^{out}$ [109]. The random walk has a stationary distribution $\boldsymbol{\pi}$, which satisfies $\sum_{u_i \in V} \boldsymbol{\pi}(i) = 1$ and

$\pi(j) = \sum_{[u_i,u_j]} \pi(i)\mathbf{P}(i,j)$ [109, 20], where $\pi(i) > 0$ for all $u_i \in V$.

To model the social network information, the basic idea is to make the latent representations of two users as close as possible if there exists a following relation between them. It can be mathematically formulated as minimizing

$$\begin{aligned} \mathcal{R}_N &= \frac{1}{2} \sum_{[u_i,u_j]\in E} \pi(i)\mathbf{P}(i,j)||\mathbf{U}_i - \mathbf{U}_j||^2 \\ &= Tr(\mathbf{U}^T(\mathbf{\Pi} - \frac{\mathbf{\Pi P} + \mathbf{P}^T\mathbf{\Pi}}{2})\mathbf{U}) \\ &= Tr(\mathbf{U}^T \triangle \mathbf{U}), \end{aligned} \tag{4.9}$$

where $\mathbf{U}_i$ denotes the low-rank representation of user u_i, $\mathbf{U}_j$ the low-rank representation of user u_j, $\triangle = \mathbf{\Pi} - \frac{\mathbf{\Pi P}+\mathbf{P}^T\mathbf{\Pi}}{2}$ is the Laplacian matrix [20], and $\mathbf{\Pi}$ denotes a diagonal matrix with $\mathbf{\Pi}(i,i) = \pi(i)$. It is straightforward to verify that the Laplacian matrix $\triangle$ has the properties introduced in Lemma (1) and Remark (1). The induction of Eq. (4.9) is straightforward and can be also found in previous work [20, 109]. This loss function will incur a penalty if two users have different low-rank representations when they have a directed relation.

Next I introduce the method to consider all of the three types of information in a general framework with the optimization algorithm.

4.4.4 Integrating Sentiment Analysis

As illustrated in Figure 4.2, I employ sentiment and network information to formulate two constraints on the matrix factorization model which is derived from content information. By considering all of the three types of information, the task of social spammer detection with sentiment information can be formulated as the following optimization problem:

$$\min_{\mathbf{U},\mathbf{V}\geq 0} \mathcal{O} = ||\mathbf{X} - \mathbf{U}\mathbf{V}^T||_F^2 + \alpha Tr(\mathbf{U}^T\mathcal{L}\mathbf{U}) + \beta Tr(\mathbf{U}^T\triangle\mathbf{U}) + \lambda(||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2), \tag{4.10}$$

where the first term is to consider content information, the second term is to introduce sentiment information, the third term is to introduce social network information, and the fourth term is for regularization to avoid overfitting. The three positive parameters α, β and λ are to control the effects of each part to the learned model.

The objective function defined in Eq. (4.10) is not convex with respect to the two variables **U** and **V** together. There is no closed-form solution for the problem. Motivated by the multiplicative and alternating updating rules discussed in [88], I now introduce an alternative algorithm to find optimal solutions for the two variables **U** and **V**. The key idea is to optimize the objective with respect to one variable, while fixing the other. The algorithm will keep updating the variables until convergence. Now I introduce the algorithm in detail.

Computation of U

Optimizing the objective function in Eq. (4.10) with respect to **U** is equivalent to solving

$$\min_{\mathbf{U}\geq 0} \quad \mathcal{O}_U = ||\mathbf{X} - \mathbf{U}\mathbf{V}^T||_F^2 + \alpha Tr(\mathbf{U}^T \mathcal{L} \mathbf{U}) + \beta Tr(\mathbf{U}^T \triangle \mathbf{U}) + \lambda ||\mathbf{U}||_F^2, \tag{4.11}$$

Let Λ_U be the Lagrange multiplier for constraint $\mathbf{U} \geq 0$, the Lagrange function $L(\mathbf{U})$ is defined as follows:

$$L(\mathbf{U}) = ||\mathbf{X} - \mathbf{U}\mathbf{V}^T||_F^2 + \alpha Tr(\mathbf{U}^T \mathcal{L} \mathbf{U}) + \beta Tr(\mathbf{U}^T \triangle \mathbf{U}) + \lambda ||\mathbf{U}||_F^2 - Tr(\Lambda_U \mathbf{U}^T), \tag{4.12}$$

By setting the derivative $\nabla_{\mathbf{U}} L(\mathbf{U}) = 0$, I get

$$\Lambda_U = -2\mathbf{X}\mathbf{V} + 2\mathbf{U}\mathbf{V}^T\mathbf{V} + 2\alpha \mathcal{L}\mathbf{U} + 2\beta \triangle \mathbf{U} + 2\lambda \mathbf{U}. \tag{4.13}$$

The Karush-Kuhn-Tucker complementary condition [12] for the nonnegativity constraint of **U** gives

$$\Lambda_U(i, j)\mathbf{U}(i, j) = 0 \; ; \tag{4.14}$$

thus, I obtain

$$[-\mathbf{X}\mathbf{V} + \mathbf{U}\mathbf{V}^T\mathbf{V} + \alpha \mathcal{L}\mathbf{U} + \beta \triangle \mathbf{U} + \lambda \mathbf{U}](i, j)\mathbf{U}(i, j) = 0. \tag{4.15}$$

Since the Laplacian matrices $\mathcal{L}$ and $\triangle$ may take any signs, I decompose it as $\mathcal{L} = \mathcal{L}^{+} - \mathcal{L}^{-}$ and $\triangle = \triangle^{+} - \triangle^{-}$. Similar to [39], it leads to the updating rule of **U**,

$$\mathbf{U}(i,j) \leftarrow \mathbf{U}(i,j)\sqrt{\frac{[\mathbf{X}\mathbf{V} + \alpha\mathcal{L}^{-}\mathbf{U} + \beta\triangle^{-}\mathbf{U}](i,j)}{[\mathbf{U}\mathbf{V}^{T}\mathbf{V} + \alpha\mathcal{L}^{+}\mathbf{U} + \beta\triangle^{+}\mathbf{U} + \lambda\mathbf{U}](i,j)}}. \tag{4.16}$$

Computation of V

Optimizing the objective function in Eq. (4.10) with respect to **V** is equivalent to solving

$$\min_{\mathbf{V}\geq 0} \ O_V = \|\mathbf{X} - \mathbf{U}\mathbf{V}^{T}\|_F^2 + \lambda\|\mathbf{V}\|_F^2, \tag{4.17}$$

Let Λ_V be the Lagrange multiplier for constraint $\mathbf{V} \geq 0$, the Lagrange function $L(\mathbf{V})$ is defined as follows:

$$L(\mathbf{V}) = \|\mathbf{X} - \mathbf{U}\mathbf{V}^{T}\|_F^2 + \lambda\|\mathbf{V}\|_F^2 - Tr(\Lambda_V\mathbf{V}^{T}), \tag{4.18}$$

By setting the derivative $\nabla_{\mathbf{V}}L(\mathbf{V}) = 0$, I get

$$\Lambda_V = -2\mathbf{X}^{T}\mathbf{U} + 2\mathbf{V}\mathbf{U}^{T}\mathbf{U} + 2\lambda\mathbf{V}. \tag{4.19}$$

The Karush-Kuhn-Tucker complementary condition [12] for the nonnegativity constraint of **U** gives

$$\Lambda_V(i,j)\mathbf{V}(i,j) = 0\ ; \tag{4.20}$$

thus, I obtain

$$[-\mathbf{X}^{T}\mathbf{U} + \mathbf{V}\mathbf{U}^{T}\mathbf{U} + \lambda\mathbf{V}](i,j)\mathbf{V}(i,j) = 0. \tag{4.21}$$

Similar to [39], it leads to the updating rule of **V**,

$$\mathbf{V}(i,j) \leftarrow \mathbf{V}(i,j)\sqrt{\frac{[\mathbf{X}^{T}\mathbf{U}](i,j)}{[\mathbf{V}\mathbf{U}^{T}\mathbf{U} + \lambda\mathbf{V}](i,j)}}. \tag{4.22}$$

The correctness and convergence of the updating rules can be proven with the standard auxiliary function approach introduced in [39, 88]. Once obtaining the low-rank user representation **U**, a supervised model can be trained based on the new latent topic space and

```
Input: $\{\mathbf{X}, \mathbf{Y}, \mathbf{G}, \alpha, \beta, \lambda, I\}$
Output: U, V, W
1 Construct matrices $\mathcal{L}$ and $\triangle$ in Eq. (4.7) and (4.9) ;
2 Initialize $\mathbf{U}, \mathbf{V} \geq 0$ ;
3 while Not convergent and $iter \leq I$ do
4     Update $\mathbf{U}(i, j) \leftarrow \mathbf{U}(i, j) \sqrt{\frac{[\mathbf{X}\mathbf{V}+\alpha\mathcal{L}^{-}\mathbf{U}+\beta\triangle^{-}\mathbf{U}](i,j)}{[\mathbf{U}\mathbf{V}^T\mathbf{V}+\alpha\mathcal{L}^{+}\mathbf{U}+\beta\triangle^{+}\mathbf{U}+\lambda\mathbf{U}](i,j)}}$ ;
5     Update $\mathbf{V}(i, j) \leftarrow \mathbf{V}(i, j) \sqrt{\frac{[\mathbf{X}^T\mathbf{U}](i,j)}{[\mathbf{V}\mathbf{U}^T\mathbf{U}+\lambda\mathbf{V}](i,j)}}$ ;
6     $iter = iter + 1$ ;
7 end
8 $\mathbf{W} = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{Y}$ ;
```

Algorithm 1: *Social Spammer Detection with Sentiment Information*

label matrix $\mathbf{Y}$. I employ the widely used Least Squares [61], which has a closed-form solution: $\mathbf{W} = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{Y}$. I present the detailed algorithm of *SDS* in Algorithm 1.

In the algorithm, I conduct initialization for Laplacian matrices, encoding matrix $\mathbf{U}$ and mixing matrix $\mathbf{V}$ from line 1 to 2. I is the number of maximum iterations. The two matrices $\mathbf{U}$ and $\mathbf{V}$ are updated with the updating rules until convergence or reaching the number of maximum iterations. The classifier $\mathbf{W}$ for social spammer detection is trained in line 8.

4.5 Experiments

In this section, I conduct extensive experiments to evaluate the effectiveness of the proposed framework *SDS*. Through the experiments, I aim to answer the following two questions,

1. How effective is the proposed framework compared with other social spammer detection methods?

2. What are the effects of the sentiment information for social spammer detection per-

Table 4.3: Social Spammer Detection Results on TUSH Dataset

	Training Data One (50%)			*Training Data Two (100%)*		
	Precision	Recall	F_1-measure (gain)	Precision	Recall	F_1-measure (gain)
Content_Net	0.893	0.924	0.908 (N.A.)	0.919	0.942	0.930 (N.A.)
Content_Lap	0.926	0.939	0.932 (+2.67%)	0.931	0.949	0.940 (+1.03%)
SMFSR	0.935	0.939	0.937 (+3.12%)	0.948	0.945	0.946 (+1.74%)
SparseSD	0.951	0.955	0.953 (+4.93%)	0.959	0.961	0.960 (+3.17%)
SDS	0.969	0.965	0.967 (+6.47%)	0.975	0.979	0.977 (+5.01%)

Table 4.4: Social Spammer Detection Results on TSS Dataset

	Training Data One (50%)			*Training Data Two (100%)*		
	Precision	Recall	F_1-measure (gain)	Precision	Recall	F_1-measure (gain)
Content_Net	0.801	0.860	0.829 (N.A.)	0.809	0.866	0.837 (N.A.)
Content_Lap	0.821	0.882	0.850 (+2.53%)	0.851	0.902	0.876 (+4.69%)
SMFSR	0.834	0.895	0.863 (+4.10%)	0.860	0.909	0.884 (+5.65%)
SparseSD	0.848	0.900	0.873 (+5.28%)	0.881	0.916	0.898 (+7.37%)
SDS	0.869	0.909	0.889 (+7.12%)	0.898	0.930	0.914 (+9.23%)

formance?

I begin by introducing the experimental setup and then compare the performance of different social spammer detection methods. Finally, I study the effects of sentiment information and the parameters on the proposed framework.

4.5.1 Experimental Settings

I follow standard experiment settings used in [105, 110] to evaluate the performance of spammer detection methods. I apply different social spammer detection methods on social media datasets. To avoid bias brought by different class distributions, the two Twitter datasets introduced in Section 4.3, TUSH and TSS, are used in the experiments. Similar to the literature, precision, recall, and F_1-measure are used as the performance metrics.

Three positive parameters are involved in the experiments, including α, β and λ in Eq. (4.10). α is to control the contribution of sentiment information, β is to control the contribution of social network information, and λ is the regularization parameter to prevent overfitting. As a common practice, all the parameters can be tuned via cross-validation with a separate validation dataset. In the experiments, I empirically set $\alpha = 0.1$, $\beta = 0.1$ and $\lambda = 0.1$ for general experiment purposes. I empirically set $k = 20$ for k-nearest neighbor defined in Eq. (4.5). The effects of the parameters on the learning model will be further discussed in Section 4.5.4.

4.5.2 Effectiveness of the Proposed Method

I now compare the proposed framework with other baseline methods, accordingly answer the first question asked above. Four baseline methods are included in the experiments:

- *Content_Net*: the content matrix **X** and adjacency matrix **G** of the social network are combined together for user representation. The basic idea here is to consider each friend of a user as a social dimension [93] for representation. I further use the widely used classifier Least Squares [32] to perform social spammer detection.

- *Content_Lap*: social network information is modeled and incorporated into a Least Squares formulation with a directed Laplacian regularization [109].

- *SMFSR*: a multi-label informed latent semantic indexing [110, 107] is used to model the content information, and undirected graph Laplacian [19] is used to incorporate the social network information. In the experiment, I convert the directed graph to an undirected one with $\mathbf{G} = max(\mathbf{G}, \mathbf{G}^T)$.
- *SparseSD*: a sparse learning framework [68] is used to model the content information, and a directed graph Laplacian [109] is used to incorporate the network information. In the experiment, the directed graph $\mathbf{G}$ is used to model social network information.
- *SDS*: the proposed framework.

Experimental results of the methods on the two Twitter datasets, THSH and TSS, are respectively reported in Table 4.3 and 4.4. In the experiment, I use five-fold cross validation for all the methods. To avoid bias brought by the sizes of the training data, I conduct two sets of experiments with different numbers of training samples. In each round of the cross validation, "Training Data One (50%)" means that I randomly chose 50% of the 80%, thus using 40% of the whole dataset for training. "Training Data One (100%)" represents that I use all the 80% data for training. Also, "gain" represents the percentage improvement of the methods in comparison with the first baseline method *Content_Net*. In the experiment, each result denotes an average of 10 test runs. By comparing the spammer detection performance of different methods, I draw the following observations:

(1) From the results in the tables, it is observed that the proposed method *SDS* consistently outperforms other baseline methods on both datasets with different sizes of training data. The proposed method achieves better results than the state-of-the-art method *SMFSR* and *SparseSD* on both datasets. I apply two-sample one-tail t-tests to compare *SDS* to the four baseline methods. The experiment results demonstrate that the proposed model performs significantly better (with significance level $\alpha = 0.01$) than the four methods.

(2) The performance of *SDS* is better than the four baselines, which are based on different strategies of utilizing content and network information. This demonstrates that the integration of sentiment information positively helps improve social spammer detection performance.

(3) Among the four baseline methods, *SMFSR* and *SparseSD* achieve better results than the first two methods *Content_Net* and *Content_Lap*. Dimensionality reduction and sparse learning methods show good performance in the studied problem. This indicates that the excellent modeling of content information significantly helps the performance of social spammer detection.

(4) The first method *Content_Net* has the worst performance among all of the four baselines. This shows that the proper use of social network information is important in social spammer detection. Simple combination of network information does not work well.

With the help of sentiment information, the proposed framework outperforms the methods incorporating content and network information. Next, I further investigate the effects of sentiment information on the social spammer detection task.

4.5.3 Effectiveness of Sentiment Information

In this subsection, I compare the effectiveness of different types of information to better understand the role of sentiment information in social spammer detection, and accordingly answer the second question asked in the beginning of this section. In particular, I compare the proposed method with the following:

- *Content*: the Least Squares is employed to train a classifier based on only content matrix **X**.
- *Network*: each friend of a user is considered as a social dimension [93] to represent the user. This is a widely used scheme in relational learning and community detection

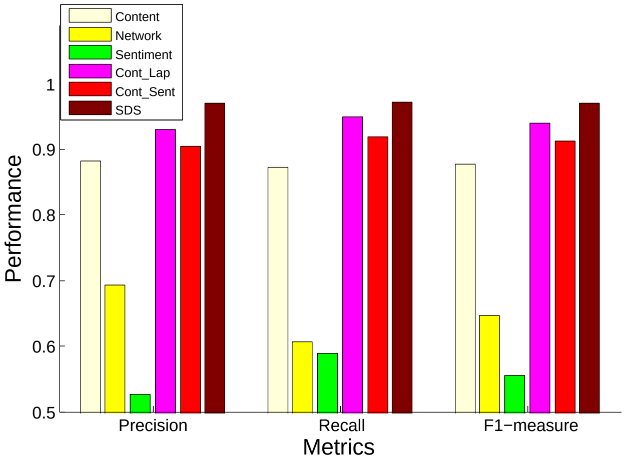


Figure 4.3: Spammer Detection Results on TUSH Dataset

for user representation. I then train a classifier based on the user-friend representation for social spammer detection.

- *Sentiment*: I first compute the sentiment score of each user and then compare its distance with the mean of spammer group and normal user group. The user is classified into the group with shorter distance.

- *Content_Lap*: the baseline is the same as that in Section 4.5.2.

- *Content_Sentiment*: sentiment information I modeled in Section 4.4.2 is combined with content information for social spammer detection.

- *SDS*: the proposed method to exploit sentiment information for social spammer detection.

The experimental results of the methods on the two datasets are respectively plotted in Figure 4.3 and 4.4. In the figures, the first five bars represent the performance of the baselines with different combinations of the information, respectively. The last bar represents the proposed method *SDS*. From the figures, I can draw the following observations:

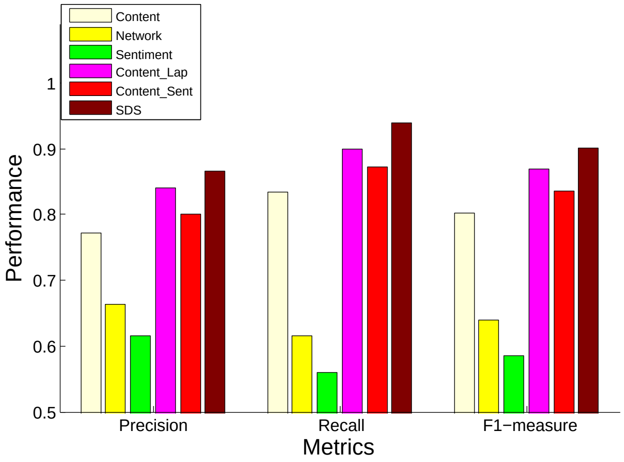


Figure 4.4: Spammer Detection Results on TSS Dataset

(1) With the integration of all the three different types of information in a unified way, the proposed framework *SDS* consistently achieves better performance than those with only content and network information. It demonstrates that the proposed method successfully makes use of useful information sources to perform effective social spammer detection.

(2) Among all of the five baseline methods, *Content_Lap* and *Content_Sentiment* achieve better performance than the first three methods. The results indicate that the integration of either network information or sentiment information into a content-based method improves the purely content-based social spammer detection performance. Comparing with traditional spammer detection methods, the use of contextual information positively helps social spammer detection performance.

(3) Among the first three methods, *Content* achieves best performance. This result has been little reported in existing work. It suggests that among the three types of information, content information is the most effective one for social spammer detection. This observation is consistent with those obtained in other platforms, such as email spam detection and Web spam detection. I can observe that *Sentiment* achieves the worst performance, which indicates that I cannot only rely one sentiment information for social spammer detection.

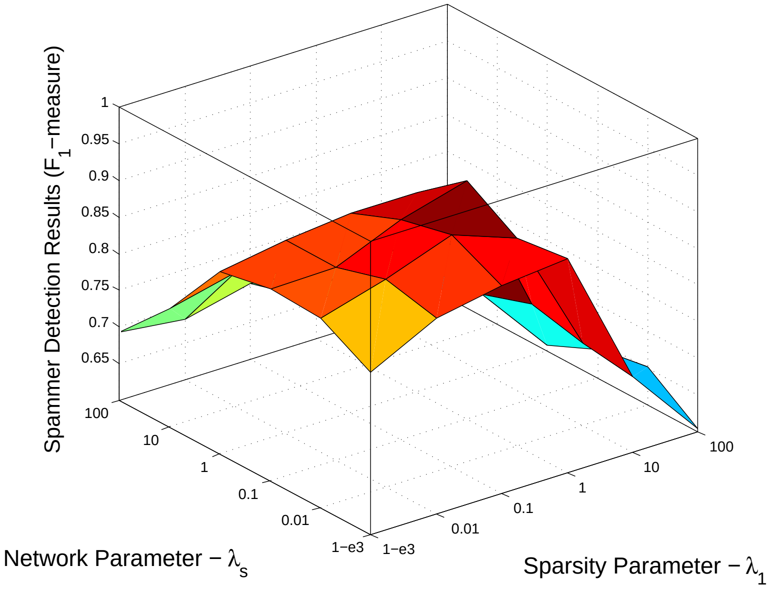


Figure 4.5: Impact of Sentiment Information (α) and Social Network Information (β) to the Proposed Framework

Although I observe that the sentiment differences do exist between spammers and normal users, sentiment information is not good enough to be an independent information source to detect spammers.

From the above discussion, it suggests that the use of sentiment information can help improve the performance of social spammer detection, although it does not work well as an independent information source. The superior performance of the proposed method *SDS* validates its excellent use of the three types of information.

4.5.4 Discussion

Two important parameters, i.e., α and β, are involved in the formulation and need to be further explored. α is to control the contribution of sentiment information, and β is to control the contribution of social network information to the model. To better understand the effects brought by the two parameters, I now conduct experiments to compare the social spammer detection performance of the proposed *SDS* on the Twitter datasets with different parameter settings.

The spammer detection results of *SDS* with different parameter settings on the TSS dataset is plotted in Figure 4.5. From the figure, I can observe that *SDS* achieves relatively good performance when $\alpha < 1$ and $\beta < 1$. When $\alpha > 1$ and $\beta > 1$, as the parameters grow, the performance of *SDS* declines. The results demonstrate that the proposed framework can achieve a relatively good performance when choosing parameter settings in a reasonable range. The performance of *SDS* is not quite sensitive to the parameters. In practice, setting α and β in [0.01, 1] achieves good performance in both datasets. Similar results can be observed on the TUSH dataset.

4.6 Summary

Social spamming has become a serious problem in almost all kinds of social media services. The distinct characteristics of social media services present new challenges for social spammer detection. Motivated by psychological findings, in this chapter, I propose to make use of sentiment information to help social spammer detection. In particular, I first conduct exploratory study on two Twitter datasets to examine the sentiment differences between spammers and normal users. The experiment results show that the sentiments posed by spammers and normal users are significantly different. The sentiment information are then modeled with a graph Laplacian and incorporated into an optimization formulation. The proposed method considers sentiment, content and network information in a unified way for social spammer detection. Extensive experiments are conducted. The experimental results demonstrate the effectiveness of the proposed framework as well as the roles of different types of information in social spammer detection.

Chapter 5

STREAMING DATA ANALYTICS FOR SOCIAL SPAMMER DETECTION

In this chapter, to handle the fast evolving social spammers, I focus on the problem of using online algorithm to update a built model. I will firstly review the background of this problem that why online learning is needed. And then I formally define the problem and introduce the proposed framework. Twitter datasets are used to evaluate the performance of the proposed method by comparing with the state-of-the-art baseline methods.

5.1 Fast-Evolving Social Spammers

Traditional spammer detection methods become less effective due to the fast evolution of social spammers. First, social spammers show dynamic content patterns in social media. Spammers' content information changes too fast to be detected by a static anti-spamming system based on offline modeling [110]. Spammers continue to change their spamming strategies and pretend to be normal users to fool the system. A built system may become less effective when the spammers create many new, evasive accounts. Second, many social media sites like Twitter have become a target of link farming [34]. The reflexive reciprocity [102, 49] indicates that many users simply follow back when they are followed by someone for the sake of courtesy. It is easier for spammers to acquire a large number of follower links in social media. Thus, with the perceived social influence, they can avoid being detected by network-based methods. Similar results targeting other platforms such as Renren [105] have been reported in literature as well. Existing systems rely on building a new model to capture newly emerging content-based and network-based patterns of social spammers. Given the rapidly evolving nature, it is necessary to have a framework that efficiently reflects the effect of newly emerging data.

Online learning has become an effective method to incrementally update existing model in large-scale data analysis. While online learning has been studied for years and shown its effectiveness in many applications such as image and video processing [71] and human computer interaction [70], it has not been applied in social spammer detection. In this chapter, I study how to capture the fast evolving nature of social spammers using online learning. In particular, I investigate:

- How do we model the content and network information in a unified framework for effective social spammer detection?
- How do we update the built model to efficiently incorporate newly emerging data objects?

My solutions to these two questions result in a new framework for Online Social Spammer Detection (*OSSD*). The proposed framework is a formulation based on directed Laplacian constrained matrix factorization, and is used to incorporate refined social network information into content modeling. Then I incrementally update the factors appropriately to reflect the rapidly evolving nature of the social spammers.

5.2 Problem Statement

In this section, I formally define the problem of incrementally update the social spammer detection model.

Let $[\mathbf{X}, \mathcal{G}, \mathbf{Y}]$ be a target social media user set with content information of social media posts $\mathbf{X}$, social network information $\mathcal{G}$, and identity label matrix $\mathbf{Y}$. I use $\mathbf{X} \in \mathbb{R}^{n \times m}$ to denote content information, i.e., messages posted by the users, where n is the number of textual features and m is the number of users. I use $\mathcal{G} = (V, E)$ to denote the social network, where nodes u and v in V represent social media users, and each directed edge $[u, v]$ in E represents a following relation from u to v. There are no self links in the graph, i.e.,

$u \neq v$. I use $\mathbf{Y} \in \mathbb{R}^{m \times c}$ to denote the identity label matrix, where c is the number of identity labels. Following literature on spammer detection [4, 63], I focus on classifying users as spammers or normal users, i.e., $c = 2$. It is straightforward to extend this setting to a multi-class classification task.

With the given notations, I formally define the problem of online social spammer detection as follows: *Given k users with their content information $\mathbf{X}^k$, social network information $\mathbf{G}^k$, and identity label information $\mathbf{Y}^k$, I learn a factorization model $\mathbf{V}^k$ and $\mathbf{U}^k$ which could be used to learn a classifier $\mathbf{W}^k$ to automatically assign identity labels for unknown users (i.e., test data) as spammers or normal users. Given one more user, the goal is to efficiently update the built model $\mathbf{V}^{k+1}$, $\mathbf{U}^{k+1}$ and $\mathbf{W}^{k+1}$ for social spammer detection based on $k + 1$ users with their content information $\mathbf{X}^{k+1}$, social network information $\mathbf{G}^{k+1}$, and identity label information $\mathbf{Y}^{k+1}$.*

5.3 Matrix Factorization for Social Spammer Detection

In this section, I propose a general framework for social spammer detection. I first discuss the modeling of content and social network information separately, and then introduce a unified framework to integrate both information.

To use content information, one way is to learn a supervised model, and apply the learned model for spammer detection. Due to the unstructured and noisy content information in social media, this method yields two problems to be directly applied to the task. First, text representation models, like n-gram model, often lead to a high-dimensional feature space because of the large size of data and vocabulary [46]. Second, In addition to the short form of texts, abbreviations and acronyms are widely used in social media, thus making the data representation very sparse.

To tackle the problems, instead of learning word-level knowledge, I propose to model the content information from topic-level. Motivated by topic modeling literature [8], a user's posts usually focus on a few topics, resulting in **X** very sparse and low-rank. The proposed method is built on a non-negative matrix factorization model (NMF) [62], which seeks a more compact but accurate low-rank representation of the users by solving the following optimization problem:

$$\min_{\mathbf{U},\mathbf{H}\geq 0} \quad ||\mathbf{X} - \mathbf{U}\mathbf{H}||_F^2, \tag{5.1}$$

where **X** is the content matrix, $\mathbf{U} \in \mathbb{R}^{n\times r}$ is a mixing matrix and $\mathbf{H} \in \mathbb{R}^{r\times m}$ with $r \ll n$ is an encoding matrix that indicates a low-rank user representation in a topic space.

Social network information has been used in many real-world applications such as sentiment analysis [89], trust prediction [90] and community deviation detection [18]. A widely used assumption is that representations of two nodes are close when they are connected with each other in the network [19, 110]. This assumption does not hold in social media. Some social media services such as microblogging have directed following relations between users. In addition, it is practical for social spammers to quickly attract a large number of followers to fool the system. Thus it is not suitable to directly apply the existing methods to the problem. Following the way used in [49] to model social network information, I employ a variant of directed graph Laplacian to model social network information. The basic idea is to make the latent representations of two users as close as possible if there exists a following relation between them. It can be mathematically formulated as minimizing

$$\begin{aligned}
\mathcal{R} = {} & \frac{1}{2}\sum_{[u,v]\in E} \boldsymbol{\pi}(u)\mathbf{P}(u,v)||\mathbf{H}_u - \mathbf{H}_v||^2 \\
= {} & tr(\mathbf{H}(\mathbf{\Pi} - \frac{\mathbf{\Pi}\mathbf{P} + \mathbf{P}^T\mathbf{\Pi}}{2})\mathbf{H}^T) \\
= {} & tr(\mathbf{H}\mathcal{L}\mathbf{H}^T), \qquad (5.2)
\end{aligned}$$

where $\mathbf{H}_u$ denotes the low-rank representation of user u, $\mathbf{H}_v$ the low-rank representation of user v, and $\mathbf{\Pi}$ denotes a diagonal matrix with $\mathbf{\Pi}(u, u) = \pi(u)$. The induction of Eq. (5.2) is straightforward and can be also found in previous work [20, 109]. This loss function will incur a penalty if two users have different low-rank representations when they have a directed relation in the graph.

I project the original content information into a latent topic space with the NMF model. By adding the network information discussed in Eq. (5.2) as a regularization, the proposed framework can be mathematically formulated as solving the following optimization problem:

$$\min_{\mathbf{H},\mathbf{U}\geq 0} \quad \mathcal{J} = ||\mathbf{X} - \mathbf{U}\mathbf{H}||_F^2 + \alpha\mathcal{R}, \tag{5.3}$$

where α is the regularization parameter to control the effects of social network information to the learned model.

It is straightforward to show that the objective function defined in Eq. (5.3) is convex of $\mathbf{U}$ and $\mathbf{H}$ separately. Following the multiplicative and alternating updating rules introduced in [88], I optimize the objective with respect to one variable, while fixing the other. Since $\mathcal{L}$ may take any signs, I decompose it as $\mathcal{L} = \mathcal{L}^+ - \mathcal{L}^-$. The updating rules for the variables are as follows:

$$\mathbf{U}(i, j) \leftarrow \mathbf{U}(i, j)\sqrt{\frac{[\mathbf{X}\mathbf{H}^T](i, j)}{[\mathbf{U}\mathbf{H}\mathbf{H}^T](i, j)}}, \tag{5.4}$$

$$\mathbf{H}(i, j) \leftarrow \mathbf{H}(i, j)\sqrt{\frac{[\mathbf{U}^T\mathbf{X} + \alpha\mathbf{H}\mathcal{L}^-](i, j)}{[\mathbf{U}^T\mathbf{U}\mathbf{H} + \alpha\mathbf{H}\mathcal{L}^+](i, j)}}. \tag{5.5}$$

It is easy to prove the correctness and convergence of the updating rules with the standard auxiliary function approach [88, 39]. Once obtaining the low-rank user representation $\mathbf{H}$, a supervised model can be trained based on the new latent topic space. I employ the widely used Least Squares [61], which has a closed-form solution: $\mathbf{W} = (\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{Y}$.

5.4 Online Learning for Social Spammer Detection

Online learning is an efficient approach to incrementally update existing model in large-scale data processing. While online learning has been widely used in various applications such as computer vision [16, 72], speech recognition [97] and bioinformatics [104], the application to spammer detection is a very new effort. In this section, I will discuss the use of online learning scheme, instead of batch-mode learning, to update the built social spammer detection model.

I have introduced a general social spammer detection model in last section. Given a model built on k users, the aim of the proposed method *OSSD* is to update factor matrices $\mathbf{U}$ and $\mathbf{H}$ by adding the $(k+1)th$ user without much computational effort. Following the formulation in Eq. (5.3), the objective function for $k+1$ users is defined as

$$\min_{\mathbf{U}^{k+1},\mathbf{H}^{k+1}\geq 0} \mathcal{J}^{k+1} = \|\mathbf{X}^{k+1} - \mathbf{U}^{k+1}\mathbf{H}^{k+1}\|_F^2 + \alpha\mathcal{R}^{k+1}, \tag{5.6}$$

where $\mathbf{X}^{k+1}$ represents the content matrix of $k+1$ users, $\mathbf{U}^{k+1}$ and $\mathbf{H}^{k+1}$ denote the factor matrices to be updated, and $\mathcal{R}^{k+1}$ indicates the objective function of graph Laplacian. This optimization problem can be solved with the batch-mode learning updating rules given by Eqs. (5.4) and (5.5). However, due to its high computational cost, an online learning updating scheme is needed.

Columns of mixing matrix $\mathbf{U}$ can be considered as the building blocks of the data, and each entity of $\mathbf{H}$ determines how the building blocks involved in the corresponding observation in $\mathbf{X}$ [44]. As the number of data objects increases, effects of each object on the representation decrease. Since the new data objects would not be able to significantly change the mixing matrix $\mathbf{U}$, it is not necessary to update the part of original encoding matrix $\mathbf{H}$ which corresponds to old objects. Thus, besides updating the mixing matrix $\mathbf{U}$, it is adequate to only update the last column of $\mathbf{H}_{k+1}$ by assuming the first k columns of $\mathbf{H}_{k+1}$ would be approximately equal to $\mathbf{H}_k$. The objective function in Eq. (5.6) can be reformulated as:

$$\begin{aligned}
\mathcal{J}^{k+1} &= ||\mathbf{X}^{k+1} - \mathbf{U}^{k+1}\mathbf{H}^{k+1}||_F^2 + \alpha \sum_{i=1}^{k+1}\sum_{j=1}^{k+1} \boldsymbol{\pi}(i)\mathbf{P}(i,j)||\mathbf{H}_i - \mathbf{H}_j||^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{k+1}(\mathbf{X}^{k+1}(i,j) - (\mathbf{U}^{k+1}\mathbf{H}^{k+1})(i,j))^2 + \alpha \sum_{i=1}^{k+1}\sum_{j=1}^{k+1} \boldsymbol{\pi}(i)\mathbf{P}(i,j)||\mathbf{H}_i - \mathbf{H}_j||^2 \\
&\approx \sum_{i=1}^{n}\sum_{j=1}^{k}(\mathbf{X}^{k}(i,j) - (\mathbf{U}^{k+1}\mathbf{H}^{k})(i,j))^2 + \sum_{i=1}^{n}(\mathbf{X}^{k+1}(i,k+1) - (\mathbf{U}^{k+1}\mathbf{H}^{k+1})(i,k+1))^2 \\
&+\alpha \sum_{i=1}^{k}\sum_{j=1}^{k} \boldsymbol{\pi}(i)\mathbf{P}(i,j)||\mathbf{H}_i - \mathbf{H}_j||^2 + 2\alpha \sum_{j=1}^{k} \boldsymbol{\pi}(k+1)\mathbf{P}(k+1,j)||\mathbf{H}_{k+1} - \mathbf{H}_j||^2,
\end{aligned}$$

and it can be further reformulated as:

$$\mathcal{J}^{k+1} \approx 2\alpha \sum_{j=1}^{k} \boldsymbol{\pi}(k+1)\mathbf{P}(k+1,j)||\mathbf{H}_{k+1} - \mathbf{H}_j||^2 + \sum_{i=1}^{n}(\mathbf{X}^{k+1}(i,k+1) - (\mathbf{U}^{k+1}\mathbf{H}^{k+1})(i,k+1))^2 + \mathcal{J}^k,$$

where $\mathcal{J}^k$ is the objective function for k users defined in Eq. (5.3). Following the updating rules introduced in [88], gradient descent optimization that yields *OSSD* is performed. When a new data object arrives, the updating rules for the variables are:

$$\mathbf{H}^{k+1}(i,k+1) \leftarrow \mathbf{H}^{k+1}(i,k+1)\sqrt{\frac{[\mathbf{A}](i,1)}{[\mathbf{B}](i,1)}},$$

$$\begin{aligned}
&\mathbf{U}^{k+1}(i,j) \leftarrow \\
&\mathbf{U}^{k+1}(i,j)\sqrt{\frac{[\mathbf{X}^k\mathbf{H}^{k^T} + \mathbf{C}](i,j)}{[\mathbf{U}^{k+1}\mathbf{H}^k\mathbf{H}^{k^T} + \mathbf{D}](i,j)}},
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{A} &= \mathbf{U}^{k+1^T}\mathbf{X}^{k+1}(*,k+1), \\
\mathbf{B} &= \mathbf{U}^{k+1^T}\mathbf{U}^{k+1}\mathbf{H}^{k+1}(*,k+1), \\
\mathbf{C} &= \mathbf{X}^{k+1}(*,k+1)\mathbf{H}^{k+1^T}(k+1,*), \\
\mathbf{D} &= \mathbf{U}^{k+1}\mathbf{H}^{k+1}(*,k+1)\mathbf{H}^{k+1^T}(k+1,*).
\end{aligned}$$

I present the algorithm of online social spammer detection in Algorithm 2. In the algorithm, I conduct initialization for the two matrices to be inferred in line 1. I is the number of maximum iterations. The two matrices are firstly learned with the method I discussed in last section, and then updated with the updating rules until convergence or reaching the number of maximum iterations from line 3 to 8. The classifier $\mathbf{W}$ is learned in line 9.

The updating rule in Eq. (5.7) is helpful in reducing the computational cost. Since $\mathbf{X}^k$ and $\mathbf{H}^k$ do not change through the learning process, instead of storing $\mathbf{X}^k$ and $\mathbf{H}^k$, there are two benefits to store results of the matrix multiplications $\mathbf{X}^k\mathbf{H}^{k^T}$ and $\mathbf{H}^k\mathbf{H}^{k^T}$. First, the dimensions of the multiplications remain the same, thus the required storage memory will be the same regardless the sizes of $\mathbf{X}^k$ and $\mathbf{H}^k$. Second, the number of matrix multiplication is the main reason of the computational complexity of traditional NMF, and it will be significantly reduced through the process with the proposed online learning scheme.

In summary, I only update columns of the encoding matrix that correspond to the new data objects in Eq. (5.7), and the updating rule in Eq. (5.7) helps in reducing the computational cost. Thus, the proposed online learning scheme is more efficient. Comparing with traditional NMF with time complexity $O(nmr^2)$, the overall time complexity of the proposed *OSSD* is $O(nr^2)$, which is independent of the number of samples m.

5.5 Experiments

In this section, I conduct extensive experiments to evaluate the effectiveness and efficiency of the proposed framework *OSSD*. Through the experiments, I aim to answer the following two questions:

1. How effective is the proposed framework compared with other methods of social spammer detection?

2. How efficient is the proposed learning framework compared with other methods?

```
**Input**: $\{\mathbf{X}, \mathbf{Y}, \mathbf{G}, \alpha, I\}$
**Output**: $\mathbf{U}, \mathbf{H}, \mathbf{W}$
Initialize $\mathbf{U}, \mathbf{H} \geq 0$
Learning $\mathbf{U}^k, \mathbf{H}^k \geq 0$
**while** *Not convergent and iter* $\leq I$ **do**
    Update $\mathbf{H}^{k+1}(i, k+1) \leftarrow$
    $\mathbf{H}^{k+1}(i, k+1) \sqrt{\frac{[\mathbf{U}^{k+1^T} \mathbf{X}^{k+1}(*, k+1)](i,1)}{[\mathbf{U}^{k+1^T} \mathbf{U}^{k+1} \mathbf{H}^{k+1}(*, k+1)](i,1)}}$
    Update $\quad \mathbf{U}^{k+1}(i, j) \leftarrow$
    $\mathbf{U}^{k+1}(i, j) \sqrt{\frac{[\mathbf{X}^k \mathbf{H}^{k^T} + \mathbf{C}](i,j)}{[\mathbf{U}^{k+1} \mathbf{H}^k \mathbf{H}^{k^T} + \mathbf{D}](i,j)}}$
    $iter = iter + 1$
**end**
$\mathbf{W} = (\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{Y}$
**return W**
```

Algorithm 2: *Online Social Spammer Detection*

5.5.1 Experimental Setup

Two Twitter datasets, **TAMU Social Honeypots Dataset (TwitterT)** and **Twitter Suspended Spammers Dataset (TwitterS)**, are used in the experiments to evaluate the effectiveness and efficiency of the proposed method. I firstly introduce the two datasets.

TwitterT[1] was originally collected from December 30, 2009 to August 2, 2010 on Twitter and introduced in [64]. It consists of Twitter users with identity labels: spammers and legitimate users. The dataset contains users, their number of followers and tweets. I filtered the non-English tweets and users with less than two tweets or two social connections. The corpus used in the study consists of 12,035 spammers and 10,912 legitimate users.

[1] `http://infolab.tamu.edu/data/`

Table 5.1: Statistics of the Datasets

	TwitterT	*TwitterS*
# of Spammers	12,035	2,049
# of Legitimate Users	10,912	11,085
# of Tweets	2,530,516	380,799
Min Degree of Users	3	3
Max Degree of Users	1,312	1,025

TwitterS: Following the data crawling process used in [105, 110], I crawled this Twitter dataset from July to September 2012 via the Twitter Search API. The users that were suspended by Twitter during this period are considered as the gold standard [94] of spammers in the experiment. I then randomly sampled the legitimate users from a publicly available Twitter dataset provided by TREC 2011.[2] According to literature [63] of spammer detection, the two classes are imbalanced, i.e., the number of legitimate users is much greater than that of spammers in the dataset. I filtered the non-English tweets and users with less than two tweets or two social connections. The statistics of the two datasets are presented in Table 5.1.

I conduct two sets of experiments for evaluation. In the first set of experiments, I follow standard experiment settings used in [4, 110] to evaluate the performance of spammer detection methods. In particular, I apply different methods on the Twitter datasets, and F_1-measure is used as the performance metric. In the second set of experiments, I compare efficiency of the proposed online learning scheme and batch-mode learning algorithms. Execution time is used as the performance metric. A standard procedure for data prepro-

[2] `http://trec.nist.gov/data/tweets/`

Table 5.2: Social Spammer Detection Results on TwitterT Dataset

	10% (gain)	25% (gain)	50% (gain)	100% (gain)
LS_Content	0.803 (N.A.)	0.829 (N.A.)	0.838 (N.A.)	0.854 (N.A.)
LS_Net	0.625 (-22.17%)	0.640 (-22.80%)	0.609 (-27.33%)	0.611 (-28.45%)
MLSI	0.865 (+7.72%)	0.882 (+6.39%)	0.873 (+4.18%)	0.896 (+4.92%)
BSSD	0.878 (+9.34%)	0.901 (+8.69%)	0.909 (+8.47%)	0.921 (+7.85%)
OSSD	0.870 (+8.34%)	0.905 (+9.17%)	0.907 (+8.23%)	0.918 (+7.49%)

Table 5.3: Social Spammer Detection Results on TwitterS Dataset

	10% (gain)	25% (gain)	50% (gain)	100% (gain)
LS_Content	0.775 (N.A.)	0.801 (N.A.)	0.811 (N.A.)	0.829 (N.A.)
LS_Net	0.603 (-22.19%)	0.610 (-23.85%)	0.612 (-24.54%)	0.597 (-27.99%)
MLSI	0.838 (+8.13%)	0.851 (+6.24%)	0.859 (+5.92%)	0.879 (+6.03%)
BSSD	0.849 (+9.55%)	0.863 (+7.74%)	0.871 (+7.40%)	0.908 (+9.53%)
OSSD	0.843 (+8.77%)	0.865 (+7.99%)	0.873 (+7.64%)	0.906 (+9.29%)

cessing is used in the experiments. I remove stop-words and perform stemming for all the tweets. The unigram model is employed to construct the feature space, tf-idf is used as the feature weight. One positive parameters α is involved in the experiments. α is to control the contribution of social network information. As a common practice, all the parameters can be tuned via cross-validation with validation data. In the experiments, I empirically set $\alpha = 0.1$ for experiments.

5.5.2 Effectiveness Evaluation

To answer the first question asked in the beginning of this section, I compare the proposed framework with following baseline methods for social spammer detection.

- *LS_Content*: the Least Squares [61] is a widely used classification method in many applications. I apply the Least Squares on the content matrix **X** for spammer detection.

- *LS_Net*: I apply the Least Squares on the adjacency matrix **G** of the social network for spammer detection.

- *MLSI*: this method considers both network and content information for spammer detection. Multi-label informed latent semantic indexing [107, 110] is used to model the content information, and undirected graph Laplacian [19] is used to incorporate the network information.

- *BSSD*: this is a variant of the proposed method. Instead of online learning, I use batch-mode learning to build the model based on the training data at one time.

- *OSSD*: the proposed online learning method.

Among the five methods, the first four are based on batch-mode learning and the last one is designed using online learning. The experimental results of the methods are summarized in Table 5.2 and 5.3. In the experiments, five-fold cross-validation is used for all the methods. To study the effects brought by different sizes of training data, I varies the training data from 10% to 100%. In particular, for each round of the experiment, 20% of the dataset is held for testing and 10% to 100% of the original training data is sampled for training. For example, “50%” indicates that I use 50% of the 80%, thus using 40% of the whole dataset for training. For *OSSD*, the online learning updates a basic model that is built based on 50% of the training data in each round. In the table, “gain” represents the percentage improvement of the methods in comparison with the first baseline method *LS_Content*. In the experiment, each result denotes an average of 10 test runs. By comparing the results of different methods on the two datasets, I draw the following observations:

(1) From the results in the tables, we can observe that our proposed methods *BSSD* and *OSSD* consistently outperform other baseline methods on both datasets with different sizes of training data. The proposed spammer detection methods achieves better results than the state-of-the-art method *MLSI* on both datasets. I apply two-sample one-tail t-tests to compare *BSSD* and *OSSD* with the three baseline methods. The experiment results demonstrate that the proposed models perform significantly better (with significance level $\alpha = 0.01$) than the three baseline methods.

(2) The last three methods achieve better results than the first two methods that are based on only one type of information. The network-based method *LS_Net* achieves the worst performance among all the methods. This demonstrates that the integration of both content and network information is helpful for effective social spammer detection.

(3) The last two methods, *OSSD* and *BSSD*, achieve comparably good performance on both datasets with different sizes of training data. This shows that, comparing with batch-mode learning method, the proposed online learning scheme does not bring in any negative effects to the accuracy of social spammer detection.

The superior performance of the proposed method answers the first question that, compared with other methods, *OSSD* is effective in spammer detection. In addition, the proposed online learning scheme can achieve comparable performance with batch-mode learning methods. Next, I evaluate efficiency of the proposed method.

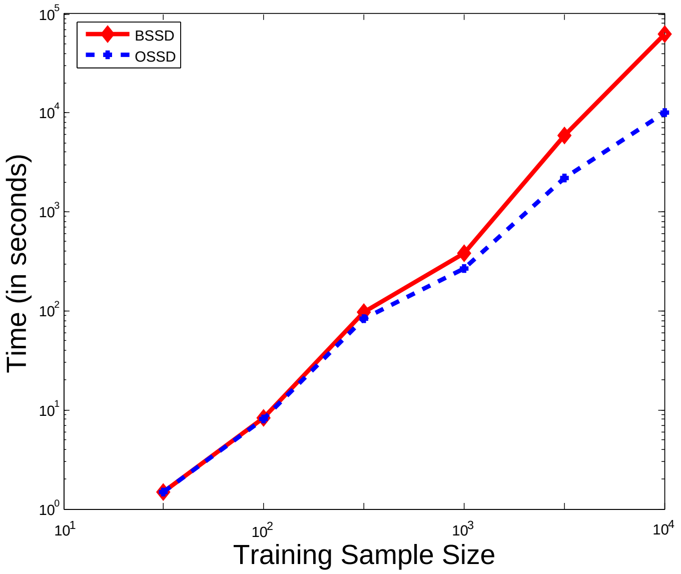


Figure 5.1: Efficiency Performance on TwitterT

5.5.3 Efficiency Evaluation

To answer the second question asked in the beginning of this section, I compare the efficiency of batch-mode learning method *BSSD* with online learning based method *OSSD*. The experiments are run on a single-CPU, eight-core 3.40Ghz machine. Experimental results of the two methods on TwitterT dataset are plotted in Figure 5.1. in the figure, x axis represents the training sample size and y axis indicates the execution time in seconds of the methods. The red curve shows the performance of *BSSD* and the blue dotted curve depicts the performance of *OSSD*.

From the figure, I observe that the online version of the proposed algorithm *OSSD* needs less running time than the batch-mode learning algorithm *BSSD*. This demonstrates that, the proposed online learning based method is more efficient than the batch-mode learning method. In many situations, especially when the training sample size is large, the differences in performance are significant between online learning and batch-mode learning method. Similar results have been observed on the TwitterS dataset. In summary, the observations answer the second question that, comparing with other methods, online learning is efficient for social spammer detection.

5.6 Summary

Social spammers are sophisticated and adaptable to game the system by continually change their content and network patterns. To handle fast evolving social spammers, I proposed to use online learning to efficiently reflect the newly emerging patterns. In this chapter, I develop a general social spammer detection framework with both content and network information, and provide its online learning updating rules. In particular, I use directed graph Laplacian to model social network information, which is further integrated into a matrix factorization framework for content information modeling. By investigating its online updating scheme, I provide an efficient way for social spammer detection. Experimental results show that the proposed method is effective and efficient comparing with other social spammer detection methods.

Chapter 6

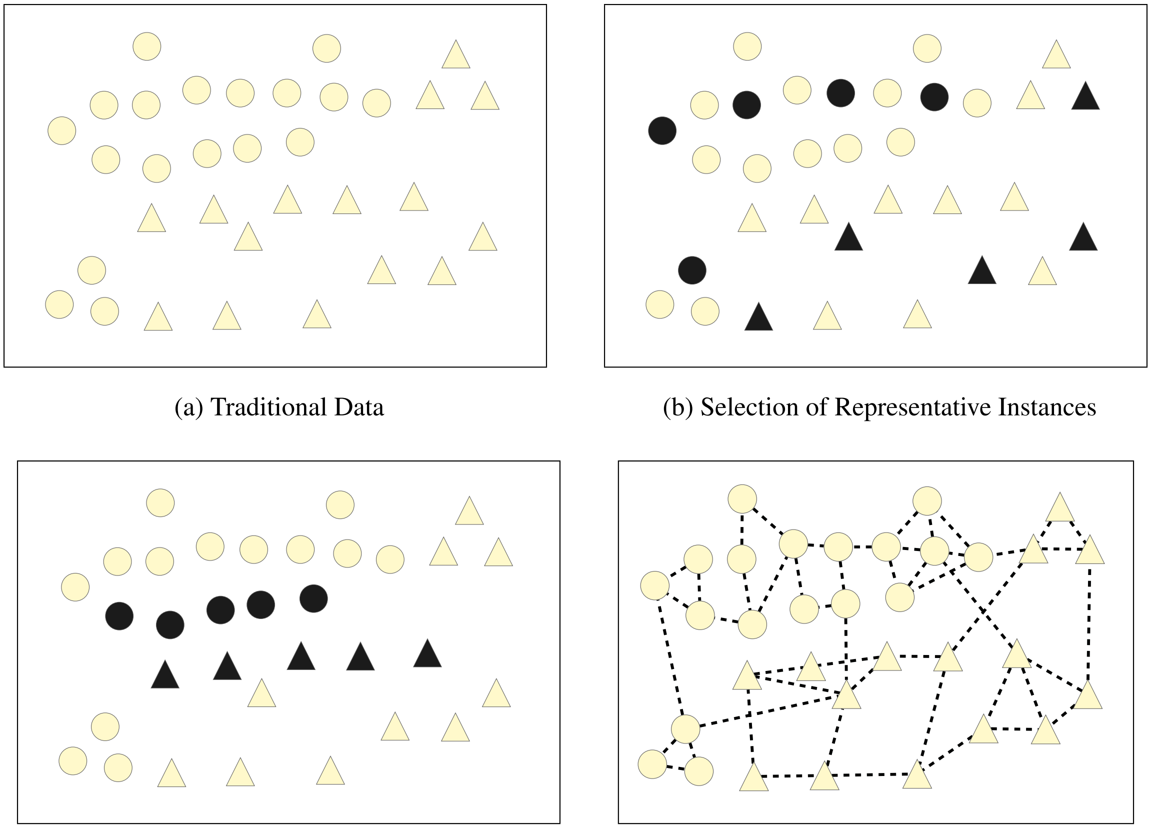
TACKLING THE LABELING BOTTLENECK: ACTIVE LEARNING FOR CONNECTED TEXTS IN SOCIAL MEDIA

As discussed in last chapters, supervised learning methods play an important role in social spammer detection and achieve very good performance. The aim is to learn a spammer detection model based on training data, which involves a basic assumption that a large number of labeled instances are available. However, labels can be expensive and time consuming to obtain in social media, which presents great challenges to the problem.

In this dissertation, I propose two ways of tackling labeling the labeling bottelneck, active learning and cross-media learning methods. In this chapter, I focus on making use of active learning method to select the most informative and representative data instances and thus tackle the labeling bottleneck and I will introduce cross-media learning in the next chapter.

6.1 Background

One effective approach to reducing the cost of labeling is *active learning* [21]. Active learning aims to determine which data instances should be selected to query for labels such that the classifier could achieve high accuracy using as few labeled instances as possible, thereby minimizing the cost of obtaining labeled data [21]. The objective of active learning is to maximize information gain given a fixed budget of labeling efforts. Active learning has been shown to be useful in many real-world applications, including graph classification [58], document classification [96], etc. However, traditional active learning methods often assume that data instances are independent and identically distributed (i.i.d.). This is not the case with social media data, in which texts are networked with each other. To the



(a) Traditional Data

(b) Selection of Representative Instances

(c) Selection of Informative Instances

(d) Networked Texts in Social Media

Figure 6.1: A Toy Example for Selecting Representative and Informative Instances in Social Media

best of my knowledge, use of active learning to handle the labeling bottleneck in networked data has not been well studied yet.

I illustrate their differences using an example in Figure 6.1. Figure 6.1a shows a binary classification example with classes represented by different shapes (circle and triangle). Traditional active learning methods select instances to label according to two main criteria, i.e., representativeness and informativeness [54]. Representativeness measures whether an instance can well represent the overall input patterns of unlabeled data, and informativeness is the ability of an instance to reduce the uncertainty of a statistical model [87]. Examples of the selection criteria are shown in Figures 6.1b and 6.1c. Unlike traditional data, as shown in Figure 6.1d, social media data provides information beyond text. A distinct feature of texts in social media is that they can be correlated through user connections, which could contain useful information that is lost in purely text-based metrics. Besides content information, relations between messages can be represented via user-message relations and user-user relations. As indicated by Figures 6.1b and 6.1c, traditional methods tend to select instances to learn the decision boundary by analyzing their content information. It necessitates investigation of active learning in handling social media messages with their relation information.

6.2 Problem Statement

Given a corpus $\mathbf{G} = (\mathbf{X}, \mathbf{S})$, where $\mathbf{X}$ is a text content matrix and $\mathbf{S}$ is a social context matrix. For the text content matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$, n is the number of messages, and m is the number of features. For the social context matrix $\mathbf{S} = (\mathbf{P}, \mathbf{F})$, $\mathbf{P} \in \mathbb{R}^{d \times n}$ is a user-message matrix, and $\mathbf{F} \in \mathbb{R}^{d \times d}$ is a user-user matrix. $\mathbf{u} = \{u_1, u_2, \ldots, u_d\}$ is the user set, where d is the number of distinct users in the corpus. In the user-message matrix, $\mathbf{P}_{ij} = 1$ denotes that message $\mathbf{t}_j$ is posted by user u_i. In the user-user friendship matrix, $\mathbf{F}_{ij} = 1$ indicates that user u_i is connected by user u_j. The graph is a directed graph, thus $\mathbf{F}$ is asymmetric.

Now I formally define active learning in social media as:

Given a corpus of social media messages $\mathbf{G}$ *with their text content information* $\mathbf{X}$*, and social context information* $\mathbf{S}$*, including the user-message matrix* $\mathbf{P}$ *and user-user matrix* $\mathbf{F}$*, and a budget B, the task is to select B instances from* $\mathbf{X}$ *to be labeled by an oracle (e.g., human annotator), so that the learned classifier* $\mathbf{W}$ *based on the labeled data can achieve maximal accuracy on unseen data (i.e., test data).*

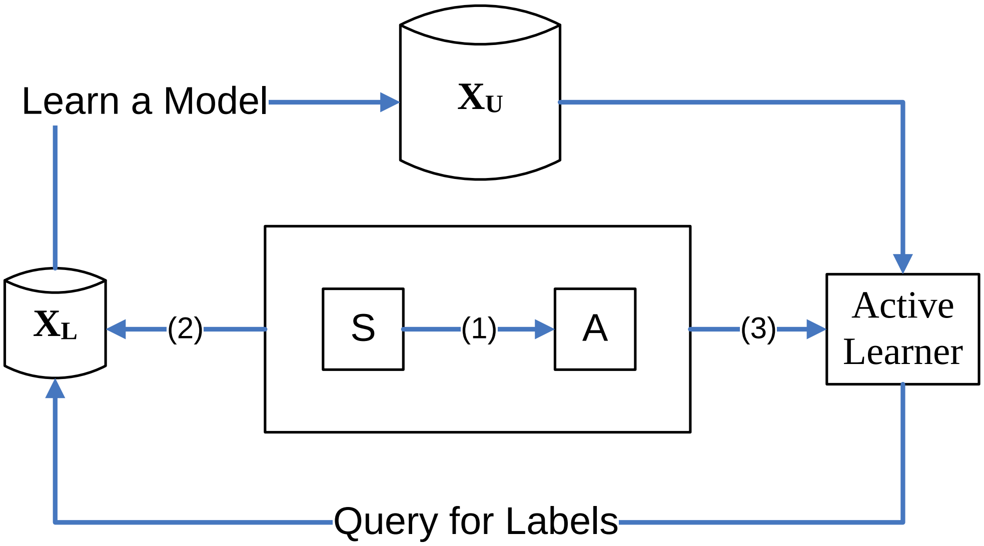


Figure 6.2: ActNeT Framework: (1) relation modeling; (2) text content modeling; (3) selection strategies for networked data.

6.3 Active Learning for Connect Texts

I plot the work flow of the proposed framework in Figure 6.2. In the figure, $\mathbf{X}_L$ represents a dataset with label information, $\mathbf{X}_U = \mathbf{X} \backslash \mathbf{X}_L$ is an unlabeled dataset, $\mathbf{S}$ is the social context matrix, and $\mathbf{A}$ is a message-message relation matrix.

In Figure 6.2, the outer cycle illustrates a traditional pool-based active learning work flow [87]. In the beginning, I have a small (or empty) labeled dataset $\mathbf{X}_L$. A learner may request labels for one or more carefully selected instances, learn from the query results, and then leverage its updated knowledge to choose instances from $\mathbf{X}_U$ to query next.

To leverage social context information, the proposed framework *ActNeT* consists of three more components than traditional active learning, as shown in the inner part of Figure 6.2. (1) In the *relation modeling* component, I extract and formally model message-message relations by analyzing social context information. (2) I incorporate the built relation information as a regularization into the *text content modeling*. (3) I studied two *selection strategies for networked data* to help the active learner choose the most representative and informative instances, in terms of network structure, to query for labels. The content modeling and relation modeling are similar to what I discussed in last section. I now elaborate the third component, the two strategies for active learning in networked data.

Traditional active learning methods select *representative* [79] or *informative* [2] instances to query for labels according to their content information only. Given the social network information available in social media data, in this section, I further explore particular features of the network topology to help select instances to query for labels.

In particular, based on the constructed message-message relation network, I examine two selection strategies for active learning.

6.3.1 Selection Strategy 1: Global Selection

As I know, representativeness-based active learning methods aim to select instances which can well represent the overall pattern of unlabeled data. For the networked data, I want to explore that whether I can select representative nodes to capture topological patterns of the whole network.

In social network analysis, many methods have been proposed to capture particular features of the network topology. The proposed methods quantify network structure with various metrics [78]. I use one of the widely used methods, PageRank [82], to select representative nodes in a network. The key idea of this selection strategy is that the nodes in the network with high PageRank scores could represent the overall patterns of the social network topology. In other words, by labeling highly representative nodes, the label information will propagate through the whole network [55].

The PageRank score can be calculated as: $\mathbf{x} = \alpha \mathbf{A} \mathbf{O}^{-1} \mathbf{x} + \beta \mathbf{1}$, where $\mathbf{x}$ is a vector of PageRank scores of all the nodes, α and β are two positive constants, $\mathbf{A}$ is the adjacency matrix, and **1** is the vector (1,1,1, ...). **O** is the diagonal matrix with elements $\mathbf{O}_{ii} = max(k_i^{out}, 1)$, and k_i^{out} is the out-degree of node i.

6.3.2 Selection Strategy 2: Local Selection

As discussed above, I select representative nodes from the whole network according to their PageRank scores. An alternative selection strategy is to consider both representativeness and informativeness of network topology in the active learning framework.

As we know, nodes sharing certain properties in a network tend to form groups with more within-group connections, which is related to a fundamental task in social network analysis – community detection [93]. Community detection algorithms aim to partition nodes in a network into different communities that have more within-group connections than between-group connections. Thus a natural choice of selecting representative nods in the network is to sample locally representative nodes from different communities. Modularity [77] is a popular community measure that explicitly takes the degree distribution into consideration and has been shown to be an effective quantity by which to measure community structure in many social network applications [24]. Here, I use modularity maximization [77] to partition the social network into communities.

After obtaining community membership information, I then select nodes with high PageRank scores in each community. I consider the messages selected from different communities as the ones that are informative in terms of network topology. The idea of this strategy is that finding locally representative nodes in each community takes both representativeness and informativeness into account.

My work focuses on studying the impact of social network information to facilitate the performance of active learning framework. It is possible to use other alternative community detection methods and network metrics in the selection procedure.

```
   Input: $\{B, b, \mathbf{X}, \mathbf{P}, \mathbf{F}, k\}$
   Output: $\mathbf{X}_L$
1  Construct Laplacian matrix $\mathcal{L}$ from $\mathbf{P}$ and $\mathbf{F}$;
2  Compute Selection Score $SS(\mathbf{x}), \mathbf{x} \in \mathbf{X}$;
3  Initialize $\mathbf{X}_L$ with $b$ instances;
4  Train $\hat{\mathbf{W}}_{LSLap}$ and $\hat{\mathbf{W}}_{Ridge}$;
5  $\mathbf{C}^k \leftarrow$ Pick $k$ instances based on $SS(\mathbf{x})$ ;
6  while $|\mathbf{X}|_L < B$ do
7      $\mathbf{x}_* = \arg\max_{\mathbf{x} \in C^k} Entropy(\mathbf{x}, \hat{\mathbf{W}}_{LSLap}, \hat{\mathbf{W}}_{Ridge})$;
8      Remove $\mathbf{x}_*$ from $\mathbf{C}^k$, add $\mathbf{x}_*$ to $\mathbf{X}_L$ ;
9      Update $\hat{\mathbf{W}}_{LSLap}$ and $\hat{\mathbf{W}}_{Ridge}$;
10 end
```

Algorithm 3: *ActNeT*: Active Learning for Networked Texts in Social Media

6.3.3 Active Learning Algorithm

By elaborating the three components plotted in Figure 6.2, here I introduce the detailed algorithm of *ActNeT* in Algorithm 3.

In line 2, I compute the selection scores $SS(x)$ for all the networked instances. The selection scores can be computed with either global or local selection strategies.

In line 3, a small number (b) of instances with highest selection scores are selected to query for labels. These instances are used to train the base learners $\hat{\mathbf{W}}_{LSLap}$ and $\hat{\mathbf{W}}_{Ridge}$ in line 4. This step presents challenges for traditional active learning method, in which they have to randomly select some instances to label as initialization. The classification result is sensitive to the initialization to some extent. As I discussed above, the two selection strategies can be applied to the readily available message-message network directly. Thus the proposed method can avoid the initialization problem.

In line 5, k instances with highest selection scores are selected from $\mathbf{X}_U$ as candidates. In lines 6 to 10, *ActNeT* proceeds in iterations until the budget B is exhausted. In each iteration, I select the most informative instances from the candidates pool $\mathbf{C}^k$ based on their vote entropy [22] evaluated by a committee of base learners. In line 7, the instance with highest entropy is defined as: $\mathbf{x}_* = \underset{\mathbf{x} \in C^k}{\arg\max} \; -\sum_{i=1}^{\mathcal{K}} \frac{V(y_i)}{\mathcal{K}} log \frac{V(y_i)}{\mathcal{K}}$, where $\mathcal{K}$ is the number of classifiers in the committee, y_i is class label provided by the ith classifier in the committee, and $V(y_i)$ is the number of occurrences of the class label y_i. In particular, I utilize LSLap and LS as base classifiers of the committee in the experiment. This step is to select the most informative nodes based on their content information. Then the selected instances are queried for labels, added to $\mathbf{X}_L$, and used to update the base classifiers.

6.4 Experiments

I present experimental results to assess the effectiveness of the proposed framework.

6.4.1 Experimental Setup

I first introduce two real-world Twitter datasets used in the experiment.

TRECTopic: Similar to experimental settings in [50, 53, 98], topics (hashtags) are considered to be class labels of tweets in the experiment. According to the topics of the tweets, I construct a ten-class Twitter dataset, which is a subset of TREC2011 data.[1] I balance the number of tweets in each class to avoid bias brought by skewed class distribution.

I further refined the tweets according to the social network information of users, which is crawled during July 2009 [60]. I filter tweets whose author has no friends or published less than two tweets. All the hashtags in the original tweets are removed during training to avoid bias brought by class labels.

TwitterStream: Following the data construction process in [50], based on the selected

[1] http://trec.nist.gov/data/tweets/

Table 6.1: Summary of Experimental Datasets

	TRECTopic	*TwitterStream*
# of Tweets	119,448	7,138
# of Unigrams	90,388	12,233
# of User	38,467	2412
# of Classes	10	10
Max Class Size	12,012	766
Min Class Size	11,885	688
Max Degree of Users	1,244	426
Min Degree of Users	1	1
Ave. Tweets per User	3.11	2.96

ten topics, tweets are crawled using Twitter Search API.[2] Tweets retrieved by the same topic are considered to be in the same category. Then I have tweets belonging to ten categories. In order to obtain the relation information, the tweets are filtered according to the same rules used in refining the TRECTopic dataset. I remove stop-words and perform stemming for all the tweets. The statistics of the two datasets are presented in Table 6.1.

In the experiment, the dataset is divided into two groups of equal size for training and testing. The active learner selects instances from the training data to query for labels. LibSVM [17] is used to train a SVM classifier based on the labeled data, and used to classify the instances in the testing data. The testing data is separate with an active learning process. Testing is done on unseen instances, but not on the remaining part of **X**. I apply different active learning methods to select B instances, and train a SVM classifier based on the selected labeled instances. Following the ratio of selection budget to the whole

[2] http://search.twitter.com/api/

data size used in active learning literature [87, 55], I set $B = 500$ for general experiment purposes. Classification accuracy is employed as the performance metric to evaluate the quality of selected instances for classification. To demonstrate the effectiveness of the proposed framework, I compare the proposed framework with following methods:

- *Random*: this method randomly selects instances to query for labels.

- *Uncertainty* [65]: the key idea of this method is to select the instances with least prediction margin between the first and second most probable class labels under the model, which is defined as: $\mathbf{x}^* = \arg\max_{x \in \mathbf{X}_U} P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x)$, where $\hat{y}_1$ and $\hat{y}_2$ are the first and second most probable labels. In this framework, the instances with small margins are considered to be ambiguous, thus knowing the true label would help the classification model discriminate more effectively between them.

- *QBC* [22]: this method selects the instances with highest disagreement level evaluated by a committee of several learners. In the experiment, entropy is used to combine the votes provided by the committee members in the experiment.

- *CLUSTER* [25]: this method samples instances with hierarchical clustering of unlabeled data.

- *ALFNET* [6]: this method clusters the nodes of a graph into several groups, and then randomly samples nodes from each cluster. The selected instances are utilized to train a collective classifier to incorporate the network information.

- *ActNeT_Global*: the proposed method with a global selection strategy.

- *ActNeT_Local*: the proposed method with a local selection strategy.

Among the baseline methods, *Random* is the way many supervised methods in social media mining used to build the training data for learning, *Uncertainty* and *QBC* are traditional content-based active learning methods, *CLUSTER* and *ALFNET* are the state-of-the-art active learning methods on content and graph information, respectively. Some methods, i.e. *Uncertainty* and *QBC*, need a small number (b) of labeled instances to train the base learners for initialization. Following experimental settings in [54, 58], I set $b = 50$, which is very small in 10-class classification tasks. Thus, 50 instances are randomly selected for initialization of the two methods in the general experiment.

There are four important parameters involved in the experiments, including λ_R, λ_L, number of communities c, and number of selected instances k. All four parameters are positive. As a common practice, λ_R and λ_L can be tuned via cross-validation. In the experiments, I set $\lambda_R = 0.005$ and $\lambda_L = 0.01$ for all the methods. I simply set $c = 10$, $k = 2 \times B$ (i.e., $k = 1000$) for general experiment purposes.

6.4.2 Effectiveness of the Proposed Method

Experimental results of the baseline methods on the two Twitter datasets are respectively plotted in Figures 6.3 and 6.4. For each classification task, I keep increasing the number of instances selected to label (budget B) from 50 to 1000, and compare the accuracy of classifiers trained based on the labeled data with different numbers of instances. From the figures, I draw the following observations:

(1) *ActNeT_Local* performs consistently better than other baselines. It demonstrates the significance of the proposed framework by exploiting the explicit network structure. *Uncertainty* and *QBC* are two classical content-based methods, and they turn out to perform similarly to each other. *ActNeT_Global* has comparable results with *ALFNET*, which further demonstrates that the representativeness and informativeness in a network are both important criteria for active learning.

(2) Specifically, the methods *ActNeT_Local*, *ActNeT_Global*, *ALFNET*, and *CLUSTER* achieve significant improvement compared with other baselines when the number of la-

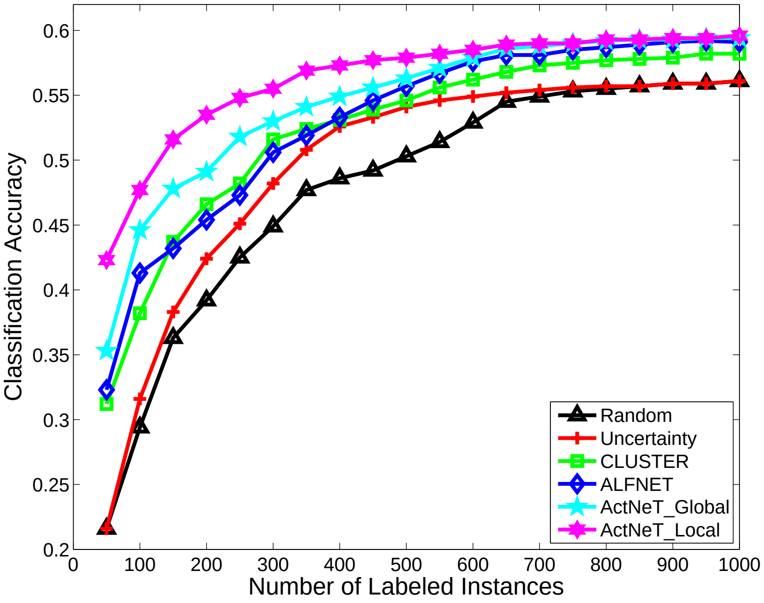


Figure 6.3: Classification Accuracy on TRECTopic

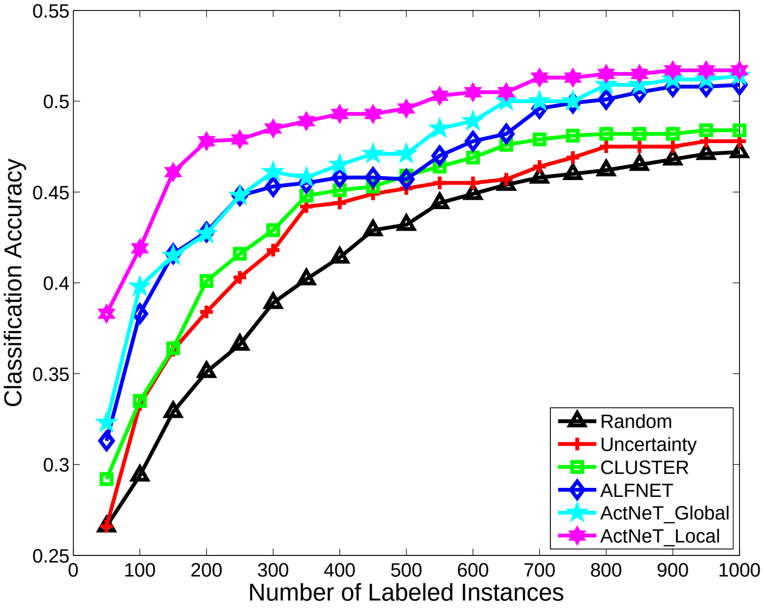


Figure 6.4: Classification Accuracy on TwitterStream

beled instances is small ($B = 50$). This is because *Uncertainty* and *QBC* randomly select a portion of data to label for training base learners. Quality of the randomly selected instances is unreliable. This property has its significance for various applications in social media when the labeling budget is small.

(3) In the figures, I do not include classification results of *WithoutAL*, which appears to be a line with fixed small number of labeled instances. Particularly, the performance of *WithoutAL* is the same as *Uncertainty* and *QBC* when their budget $B = 50$. The unsatisfactory accuracy suggests that more labeled training instances selected by different active learning methods are necessary for classification.

6.5 Summary

In this chapter, I develop a novel active learning framework to handle the networked texts in social media. In particular, I extract relations between texts based on social theories, and model the relations using graph Laplacian, which is employed as a regularization to ridge regression. Thus the relations between messages can be naturally embedded into the active learning process to effectively select informative instances from the data. I further propose global and local selection strategies for networked instances. Experimental results show that message-message relations are helpful for active learning on social media messages. Empirical evaluations demonstrate that the framework *ActNeT_Local*, which considers representativeness and informativeness in active learning, significantly outperforms the representative baselines on two real-world datasets.

Chapter 7

TACKLING THE LABELING BOTTLENECK: CROSS-MEDIA LEARNING FOR SOCIAL SPAMMER DETECTION

I introduced to make use of active learning method to tackle the labeling bottleneck in last chapter. Active learning method is effective when we want to directly learn a model based on training data from the same domain. However, in many real-world applications, it is possible for us to obtain information from heterogeneous data domains. In this chapter, I focus on the problem of learning cross-media information for social spammer detection. I will firstly review the background of this problem that why label information is essential. And I conduct data analysis to explore the possibility of using cross-media knowledge. Then I formally define the problem and introduce the proposed algorithm. Real-world Twitter datasets are used to evaluate the performance of the proposed method by comparing with the state-of-the-art baseline methods.

7.1 Background

As I discussed, existing social spammer detection methods can roughly be divided to two categories, content-based methods and network-based methods. A straightforward way to perform content-based spammer detection [63] is to model this task as a supervised learning problem. These methods extract effective textual features from the messages and build a classifier or a regressor based on the features. Given a new user, the built model can output a class label or score to determine whether it is a spammer based on microblogging messages the user posted. Content-based methods become difficult to be directly applied due to the distinct features of microblogging data. First, in microblogging, it is time-consuming and labor intensive to obtain labeled data, which is essential in building an effective supervised spammer detection model. Given the size and dynamic nature of microblogging, a manual labeling process is neither scalable nor sensible. Second, the texts in microblogging are short and noisy; thus, it lacks sufficient aggregated information to evaluate the given messages. These present great challenges to directly making use of existing content-based methods for effective spammer detection in microblogging.

While the problem of spamming in microblogging is relatively new, it has been extensively studied for years in other platforms, e.g., email communication [7], SMS [36] and the web [100]. Similarly, the spammers in these platforms unfairly overwhelm other users by spreading unwanted information, which leads to phishing, malware, and scams [49]. Also, it has been reported in Natural Language Processing (NLP) literature that microblogging is not as noisy as was expected [3]. Although microblogging is an informal communication medium, it has been shown to be similar to other platforms [52] and it is seemingly possible to employ NLP tools to “clean” it [28]. Motivated by the previous findings, I explore the possibility of using knowledge learned from other platforms to facilitate spammer detection in the context of microblogging.

In this chapter, I explore the use of resources available in other media to help spammer detection in microblogging. To study this problem, I am particularly interested in answering the following questions:

- Are the resources from other media potentially helpful for spammer detection in microblogging?
- How do we explicitly model and make use of the resources from other media for spammer detection?
- Is the knowledge learned from other cross-media resources really helpful for microblogging spammer detection?

Table 7.1: Statistics of the Datasets

	TweetH	*TweetS*	*Email*	*SMS*	*Web*
# of Spam Messages	1,310,318	71,842	10,582	747	22,386
# of Legitimate Messages	1,220,198	308,957	13,990	4827	N.A.
# of Messages	2,530,516	380,799	24,572	5574	82,386
Avg. # of Words per Document	18.64	17.88	168.87	14.59	57.67

Specifically, in this chapter, I firstly conduct a quantitative analysis of linguistic variation of spam resources from different media, and formally define the problem of leveraging knowledge across media for spammer detection in microblogging. Then I present a novel framework of leveraging knowledge from existing corpora to help spammer detection in microblogging. Through experiments on real-world datasets, I demonstrated the effectiveness of the proposed framework.

7.2 Comparing Linguistic Styles

This work is motivated by numerous spam resources available in other well-studied media, e.g., email, SMS and web. A natural question could be, given the short and noisy form of microblogging messages, how different are the texts in microblogging when compared to those in other media? Before proceeding further, I also examine whether the textual information from other media is potentially useful in the problem I study.

7.2.1 Experiment Preparation

Two Twitter datasets are used in the study for experiment purposes, i.e., TAMU Social Honeypots and Twitter Suspended Spammers. In addition, three representative datasets from different types of media, including Enron Email Dataset, SMS Dataset and Web

Dataset, are used in the analysis. The statistics of the datasets are presented in Table 7.1. Now I introduce the datasets in detail.

TAMU Social Honeypots Dataset (TweetH): Lee *et al.* [63] created a collection of 41,499 Twitter users with identity labels: spammers and legitimate users. The dataset was collected from December 30, 2009 to August 2, 2010 on Twitter. It consists of users, their number of followers and tweets. I filtered the non-English tweets and users with less than two tweets.

Twitter Suspended Spammers Dataset (TweetS): I employed a data crawling process, which is similar to [105, 110], to construct this dataset. I first crawled a Twitter dataset from July to September 2012 via the Twitter Search API. The users that were suspended by Twitter during this period are considered as the gold standard [105] of spammers in the experiment. I then randomly sampled the legitimate users from a publicly available Twitter dataset provided by TREC 2011.[1] I filtered the non-English tweets and users with less than two tweets.

The first dataset TweetH has balanced number of spammers and legitimate users. To avoid effects brought by different class distribution, according to the literature of spammer detection [63], I made the two classes in TweetS imbalanced, i.e., the number of legitimate users is much greater than that of spammers in the dataset.

Enron Email Dataset (Email): In the experiment, I used a subset of a widely used Enron email dataset,[2] which is collected during the investigation of Enron corporation and contains more than 200,000 emails between its employees. The emails in this dataset are preprocessed and used as a testbed in [75] for experiments. Each email in the dataset is labeled as either "spam" or "ham".

SMS Dataset(SMS): I used the SMS spam collection provided by Almeida *et al.* [1]

[1] `http://trec.nist.gov/data/tweets/`

[2] `http://www.isi.edu/~adibi/Enron/Enron.htm`

for analysis. This dataset is constructed based on two sources, Grumbletext web site[3] and NUS SMS Corpus.[4] The spam messages were manually labeled, and the ham messages were randomly sampled from the NUS SMS Corpus. To the best of my knowledge, this is the largest public SMS spam dataset.

Web Dataset (Web): Web spam is a key challenge for internet users. Web pages which are created to deceive other users by manipulating search engine. Webb *et al.* [100] constructed the Web Dataset. This is the largest publicly available dataset to the best of my knowledge. I removed the web pages that have no textual content or only contain http request error information.

7.2.2 Linguistic Metrics

Many metrics have been proposed in literature of Natural Language Processing and communication [3, 99] to evaluate the style of a language. In this subsection, I briefly introduce the metrics used in the study.

Basic Statistics: average Word Length (**WL**, in characters) and average Sentence Length (**SL**, in words) are used to evaluate the basic style of different datasets. In addition to those, I further employ other widely used lexical metrics in the analysis. I list the metrics below.

Type-Token Ratio (TTR): This is a widely used metric to evaluate the difficulty (or readability) of words, sentences and documents by measuring their lexical variety [14, 106]. The basic assumption of using TTR is that difficult words are those that appear least often in a document. Given a corpus D, TTR is calculated as $TTR(D) = \sum_{w \in D} \frac{Freq(w)}{Size(D)}$, where w means a word (token) in the corpus, $Freq(w)$ means word frequency of w in D, and $Size(D)$ means the number of distinct words (types) in D. In practice, a higher TTR

[3] `http://www.grumbletext.co.uk/`

[4] `http://wing.comp.nus.edu.sg/SMSCorpus/`

Table 7.2: Lexical Analysis Results

	Basics		*Lexical Analysis*		
	WL	SL	TTR	LD	OOV
TweetH	4.12	12.95	5.42	0.48	0.32
TweetS	3.95	12.38	5.65	0.50	0.31
Email	4.52	17.88	5.46	0.53	0.29
SMS	3.99	12.60	6.54	0.45	0.34
Web	4.81	18.66	6.13	0.48	0.32

indicates a larger amount of lexical variation and a lower score indicates relatively less lexical variation [106].

Lexical Density (LD): I employ lexical density to further analyze the stylistic difference between different corpora. Lexical words [41], also known as content or information carrying words, refer to verbs, nouns, adjectives and adverbs. Similarly, given a document D, LD is defined as $LD(D) = \sum_{w \in Lex} \frac{Freq(w)}{Size(D)}$, where Lex means the whole lexical words dictionary. In general, a higher lexical density indicates that it is a more formal document, and a lower lexical density represents a more conversational one.

Out-of-Vocabulary (OOV): This metric is to measure the ratio of out-of-vocabulary words in the corpora. I use a list of top 10,000 words with highest frequency provided by the Project Gutenberg [42] in the study. In general, a higher OOV rate indicates that the language is more informal. Many NLP and IR models suffer from high OOV rates.

7.2.3 Linguistic Variation Analysis

In this subsection, I introduce the lexical analysis results on the datasets from different media.

Experimental results of the lexical analysis are presented in Table 7.2. By comparing the results of different metrics, I observe the following: (1) The word lengths of different corpora are very similar, and the sentence lengths of TweetH, TweetS and SMS are smaller than those of more formal media Email and Web. This indicates that the textual form of microblogging data is similar to SMS, and relatively different from email and web. (2) In most of the tests, microblogging data is similar to the datasets from the other media. It demonstrates that, although microblogging is considered an informal media, the language use is similar to that in other media, especially in email and SMS. I observe that the type-token ratios of microblogging are smaller than those of SMS and web. It suggests that the language used in microblogging is easier than that in the other two platforms.

I further employ hypothesis testing to examine the lexical differences between microblogging datasets and other datasets. For each lexical metric, I form a null hypothesis for a microblogging dataset and a dataset from the other media. The null hypothesis is: in terms of the specific lexical metric, there is no difference between microblogging data and data from the other media. I test the hypotheses on all pairs of the datasets for all the three lexical metrics.

In particular, to verify the difference between TweetH and Email datasets on the TTR, I construct two vectors $\mathbf{ttr}_{th}$ and $\mathbf{ttr}_{em}$. Each element of the first vector $\mathbf{ttr}_{th}$ is obtained by calculating the TTR score of a subset sampled with bootstrapping from TweetH dataset. Similarly, each element in the second vector corresponds to the TTR score of a subset sampled with bootstrapping from Email dataset. In the experiment, the two vectors contain

Table 7.3: Hypothesis Testing Results (P-Values)

	TweetH			*TweetS*		
	TTR	LD	OOV	TTR	LD	OOV
Email	0.318	0.108	0.442	0.234	0.267	0.308
SMS	<0.01	0.205	0.350	<0.01	0.082	0.163
Web	<0.01	0.623	0.398	0.108	0.551	0.462

equal number of elements.[5] Each element in the vectors corresponds to 100 data instances. I formulate a two-sample two-tail t-test on the two constructed vectors $\mathbf{ttr}_{th}$ and $\mathbf{ttr}_{em}$. I examine whether there is sufficient statistical evidence to support the hypothesis that the two datasets have the same sample mean, and it is defined as follows:

$$\begin{aligned} H_0 &: \mu_{th} - \mu_{em} = 0 \\ H_1 &: \mu_{th} - \mu_{em} \neq 0 \end{aligned} \tag{7.1}$$

where H_0 is the null hypothesis, H_1 is the alternative hypothesis, and μ_c and μ_r represent the sample means of the two vectors, respectively. Similarly, I form the hypothesis testings for other pairs of datasets with other lexical metrics.

The hypothesis testing results, p-values, are summarized in Table 7.3. From the table, I can observe the following:

(1) With few exceptions, the results are much greater than the significance level $\alpha = 0.05$. It demonstrates that there is no statistical evidence to reject the null hypothesis in the tests on the two datasets. In other words, the results suggest that microblogging data is not significantly different from the datasets in other media.

[5] Note this is the setting used for experiment purposes, and it is not a mandatory setting for a two-sample t-test.

(2) In some of the tests, microblogging data appears more similar to Email than the other datasets.

While characteristics of different datasets appear different, there are no statistically significant lexical differences between them. The resources from other media are potentially useful in the task I study. Next, I formally define the problem I study and introduce the proposed learning framework for spammer detection.

7.3 Problem Statement

In this section, I formally define the problem of cross-media learning.

Let $\mathbf{S} = [\mathbf{X}, \mathbf{Y}]$ be available resources from other media, with the content information $\mathbf{X}$ and identity label matrix $\mathbf{Y}$. I use term-user matrix $\mathbf{X} \in \mathbb{R}^{m \times d}$ to denote content information, i.e., posts written by the users, where m is the number of textual features, and d is the number of users in the other media. $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_r\}$ means the combination of content information from multiple media, and $\mathbf{Y} \in \mathbb{R}^{d \times c} = \{\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_r\}$ means the combination of label information from the media. For each user $(\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{R}^{m+c}$ consists of message content and identity label, where $\mathbf{x}_i \in \mathbb{R}^m$ is the message feature vector and $\mathbf{y}_i \in \mathbb{R}^c$ is the spammer label vector. In this chapter, I consider the task I study as a two-class classification problem, i.e., $c = 2$. For example, $\mathbf{y}_i = (1, 0)$ means this user is a spammer. $\mathbf{y}_i^T \mathbf{y}_i = 1$ constrains that $\mathbf{y}_i$ has to have one label and cannot be $(0, 0)$ or $(1, 1)$. It is practical to extend this setting to a multi-class or regression problem. I use $\mathbf{T} \in \mathbb{R}^{m \times n}$ to denote the content information of microblogging users, where m is the number of textual features, and n is the number of users in microblogging. The texts from microblogging and other media share the same feature space.

I now formally define the problem: *There are a set of resources* $\mathbf{S}$ *from different media, with the content information* $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_r\}$ *and identity label information* $\mathbf{Y} = \{\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_r\}$. *Given the content information* $\mathbf{T}$ *from microblogging, the goal is to*

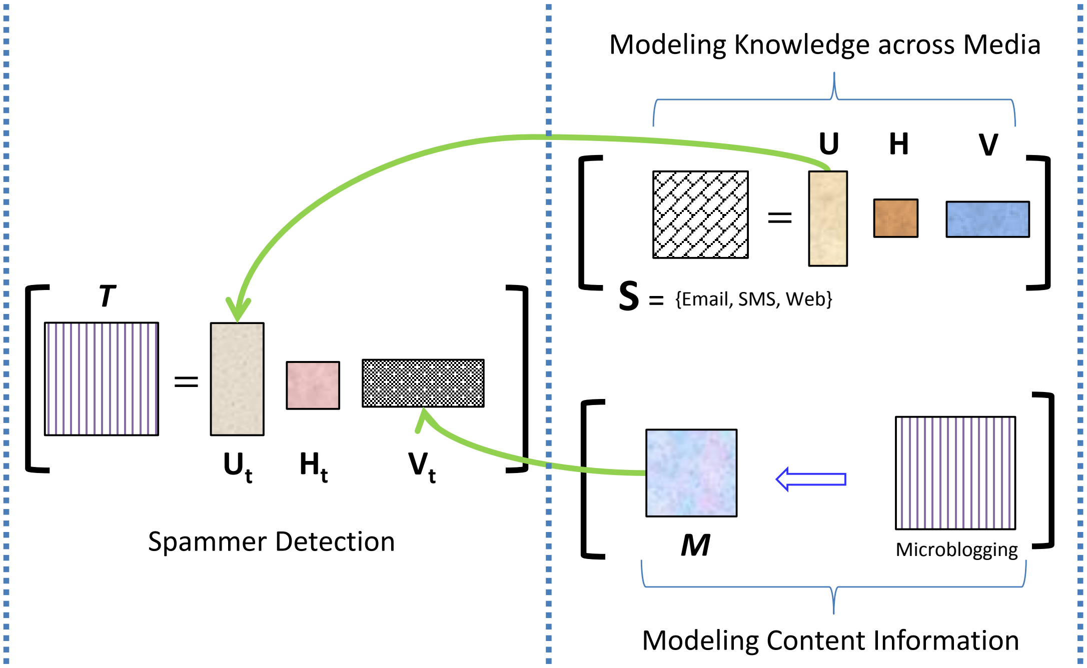


Figure 7.1: Illustration of the Proposed Spammer Detection Framework

infer the identity labels for unknown users in **T** *as spammers or legitimate users.*

7.4 Cross-Media Learning for Spammer Detection

I plot the work flow of the proposed framework in Figure 7.1. From the figure, there are two constraints on the learned model for spammer detection. As shown in the upper right part of the figure, the first constraint is from the lexicon information **U**, which is learned from the other media sources **S**. As shown in the lower right part of the figure, the second constraint is a Laplacian regularization **M** learned from microblogging content information. I now introduce each part of the proposed framework in detail.

7.4.1 Learning Knowledge from Cross-Media Resources

As I discussed in the last section, from a linguistic perspective, it does not show significant difference between microblogging data and other types of data. A straightforward method to make use of external information is to learn a supervised model based on data from the other media, and apply the learned classifier on microblogging data for spammer detection. However, this method yields two problems to be directly applied to the task. First, text representation models, like n-gram model, often lead to a high-dimensional feature space because of the large size of data and vocabulary. Second, texts in the media are short, thus making the data representation very sparse [52].

To tackle the problems, instead of learning knowledge at word-level, I propose to capture the external knowledge from topic-level. In particular, the proposed method is built on the orthogonal nonnegative matrix tri-factorization model (ONMTF) [26]. The basic idea of the ONMTF model is to cluster data instances based on distribution of features, and cluster features according to the distribution of data instances. The principle of ONMTF is consistent with PLSI [43], in which each document is a mixture of latent topics that each word can be generated from. The ONMTF can be formulated by optimizing:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{H},\mathbf{V}\geq 0} \quad ||\mathbf{X} - \mathbf{U}\mathbf{H}\mathbf{V}^T||_F^2, \\ &s.t. \;\; \mathbf{U}^T\mathbf{U} = \mathbf{I}, \;\; \mathbf{V}^T\mathbf{V} = \mathbf{I}, \end{aligned} \tag{7.2}$$

where $\mathbf{X}$ is the content matrix, and $\mathbf{U} \in \mathbb{R}_+^{m\times c}$ and $\mathbf{V} \in \mathbb{R}_+^{d\times c}$ are nonnegative matrices indicating low-dimensional representations of words and users, respectively. m is the size of vocabulary, c is the number of classes, d is the number of users. $\mathbf{H} \in \mathbb{R}_+^{c\times c}$ provides a condensed view of $\mathbf{X}$. The orthogonal and nonnegative conditions of $\mathbf{U}$ and $\mathbf{V}$ provide a hard assignment of class label to the words and users.

With the ONMTF model, the original content information is projected from the other media into a latent topic space. By adding a topic-level least squares penalty to the ON-

MTF, the proposed framework can be mathematically formulated as solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{U},\mathbf{H},\mathbf{V},\mathbf{W}\geq 0} \quad & \mathcal{J} = ||\mathbf{X} - \mathbf{U}\mathbf{H}\mathbf{V}^T||_F^2 + \lambda||\mathbf{V}\mathbf{W} - \mathbf{Y}||_F^2, \\ & s.t. \;\; \mathbf{U}^T\mathbf{U} = \mathbf{I}, \;\; \mathbf{V}^T\mathbf{V} = \mathbf{I}, \end{aligned} \tag{7.3}$$

where **W** represents the weights and **Y** is the label matrix. In the formulation, the first term is the basic factorization model, and the second introduces label information from the other media by using a linear penalty. λ is to control the effect of external information to the learned lexicon **U**, in which each row represents the predicted label of a word.

As the problem in Eq. (7.3) is not convex with respect to the four variables together, there is no closed-form solution for the problem. Next, I introduce an alternative scheme to solve the optimization problem.

Optimization Algorithm

Following [26], I propose to optimize the objective with respect to one variable, while fixing others. The algorithm will keep updating the variables until convergence.

Computation of H: Optimizing the objective function in Eq. (7.3) with respect to **H** is equivalent to solving

$$\min_{\mathbf{H}\geq 0} \quad \mathcal{J}_H = ||\mathbf{X} - \mathbf{U}\mathbf{H}\mathbf{V}^T||_F^2. \tag{7.4}$$

Let Λ_H be the Lagrange multiplier for constraint $\mathbf{H} \geq 0$; the Lagrange function $L(\mathbf{H})$ is defined as follows:

$$L(\mathbf{H}) = ||\mathbf{X} - \mathbf{U}\mathbf{H}\mathbf{V}^T||_F^2 - Tr(\Lambda_H \mathbf{H}^T). \tag{7.5}$$

By setting the derivative $\nabla_{\mathbf{H}} L(\mathbf{H}) = 0$, I get

$$\Lambda_H = -2\mathbf{U}^T\mathbf{X}\mathbf{V} + 2\mathbf{U}^T\mathbf{U}\mathbf{H}\mathbf{V}^T\mathbf{V}. \tag{7.6}$$

The Karush-Kuhn-Tucker complementary condition [12] for the nonnegativity constraint of $\mathbf{H}$ gives

$$\Lambda_H(i, j)\mathbf{H}(i, j) = 0 \; ; \tag{7.7}$$

thus, I obtain

$$[-\mathbf{U}^T\mathbf{X}\mathbf{V} + \mathbf{U}^T\mathbf{U}\mathbf{H}\mathbf{V}^T\mathbf{V}](i, j)\mathbf{H}(i, j) = 0. \tag{7.8}$$

Similar to [26], it leads to the updating rule of $\mathbf{H}$,

$$\mathbf{H}(i, j) \leftarrow \mathbf{H}(i, j)\sqrt{\frac{[\mathbf{U}^T\mathbf{X}\mathbf{V}](i, j)}{[\mathbf{U}^T\mathbf{U}\mathbf{H}\mathbf{V}^T\mathbf{V}](i, j)}}. \tag{7.9}$$

Computation of U: Optimizing the objective function in Eq. (7.3) with respect to $\mathbf{U}$ is equivalent to solving

$$\begin{aligned} \min_{\mathbf{U}\geq 0} \quad & \mathcal{J}_U = \|\mathbf{X} - \mathbf{U}\mathbf{H}\mathbf{V}^T\|_F^2 \\ s.t. \quad & \mathbf{U}^T\mathbf{U} = \mathbf{I}. \end{aligned} \tag{7.10}$$

Let Λ_U and Γ_U be the Lagrange multipliers for constraints $\mathbf{U} \geq 0$ and $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, respectively; the Lagrange function $L(\mathbf{U})$ is defined as follows:

$$L(\mathbf{U}) = \|\mathbf{X} - \mathbf{U}\mathbf{H}\mathbf{V}^T\|_F^2 - Tr(\Lambda_U\mathbf{U}^T) + Tr(\Gamma_U(\mathbf{U}^T\mathbf{U} - \mathbf{I})) \tag{7.11}$$

By setting the derivative $\nabla_{\mathbf{U}}L(\mathbf{U}) = 0$, I get

$$\Lambda_U = -2\mathbf{X}\mathbf{V}\mathbf{H}^T + 2\mathbf{U}\mathbf{H}\mathbf{V}^T\mathbf{V}\mathbf{H}^T + 2\mathbf{U}\Gamma_U. \tag{7.12}$$

With the KKT complementary condition for the nonnegativity constraint of $\mathbf{U}$, I have

$$\Lambda_U(i, j)\mathbf{U}(i, j) = 0; \tag{7.13}$$

thus, I obtain

$$[-\mathbf{X}\mathbf{V}\mathbf{H}^T + \mathbf{U}\mathbf{H}\mathbf{V}^T\mathbf{V}\mathbf{H}^T + \mathbf{U}\Gamma_U](i, j)\mathbf{U}(i, j) = 0, \tag{7.14}$$

where

$$\Gamma_U = \mathbf{U}^T\mathbf{X}\mathbf{V}\mathbf{H}^T - \mathbf{H}\mathbf{V}^T\mathbf{V}\mathbf{H}^T. \tag{7.15}$$

Let $\Gamma_U = \Gamma_U^+ - \Gamma_U^-$, where $\Gamma_U^+(i,j) = (|\Gamma_U(i,j)| + \Gamma_U(i,j))/2$ and $\Gamma_U^-(i,j) = (|\Gamma_U(i,j)| - \Gamma_U(i,j))/2$ [26]; I get

$$[-(\mathbf{X}\mathbf{V}\mathbf{H}^T + \mathbf{U}\Gamma_U^-) + (\mathbf{U}\mathbf{H}\mathbf{V}^T\mathbf{V}\mathbf{H}^T + \mathbf{U}\Gamma_U^+)](i,j)\mathbf{U}(i,j) = 0, \tag{7.16}$$

which leads to the updating rule of **U**,

$$\mathbf{U}(i,j) \leftarrow \mathbf{U}(i,j)\sqrt{\frac{[\mathbf{X}\mathbf{V}\mathbf{H}^T + \mathbf{U}\Gamma_U^-](i,j)}{[\mathbf{U}\mathbf{H}\mathbf{V}^T\mathbf{V}\mathbf{H}^T + \mathbf{U}\Gamma_U^+](i,j)}}. \tag{7.17}$$

Computation of V: Optimizing the objective function in Eq. (7.3) with respect to **V** is equivalent to solving

$$\begin{gathered}\min_{\mathbf{V}\geq 0} \ \mathcal{J} = \|\mathbf{X} - \mathbf{U}\mathbf{H}\mathbf{V}^T\|_F^2 + \lambda\|\mathbf{V}\mathbf{W} - \mathbf{Y}\|_F^2 \\ s.t. \ \ \mathbf{V}^T\mathbf{V} = \mathbf{I}.\end{gathered} \tag{7.18}$$

Similar to the computation of **U**, by introducing two Lagrange multipliers Λ_V and Γ_V for the constraints, I get

$$[-(\mathbf{X}^T\mathbf{U}\mathbf{H} + \lambda\mathbf{Y}\mathbf{W}^T + \mathbf{V}\Gamma_V^-) + (\mathbf{V}\mathbf{H}^T\mathbf{U}^T\mathbf{U}\mathbf{H} + \lambda\mathbf{V}\mathbf{W}\mathbf{W}^T + \mathbf{V}\Gamma_V^+)](i,j)V(i,j) = 0, \tag{7.19}$$

which leads to the updating rule of **V**,

$$\mathbf{V}(i,j) \leftarrow \mathbf{V}(i,j)\sqrt{\frac{[\mathbf{X}^T\mathbf{U}\mathbf{H} + \lambda\mathbf{Y}\mathbf{W}^T + \mathbf{V}\Gamma_V^-](i,j)}{[\mathbf{V}\mathbf{H}^T\mathbf{U}^T\mathbf{U}\mathbf{H} + \lambda\mathbf{V}\mathbf{W}\mathbf{W}^T + \mathbf{V}\Gamma_V^+](i,j)}} \tag{7.20}$$

Computation of W: Optimizing the objective function in Eq. (7.3) with respect to **W** is equivalent to solving

$$\min_{\mathbf{W}\geq 0} \ \mathcal{J} = \|\mathbf{V}\mathbf{W} - \mathbf{Y}\|_F^2. \tag{7.21}$$

Similar to the computation of **U**, by introducing a Lagrange multiplier and satisfying KKT condition, I obtain

$$[\mathbf{V}^T\mathbf{V}\mathbf{W} - \mathbf{V}^T\mathbf{Y}](i,j)\mathbf{W}(i,j) = 0, \tag{7.22}$$

```
Input: $\{\mathbf{X}, \mathbf{Y}, \lambda, I\}$
Output: $\mathbf{V}$
1 Initialize $\mathbf{U}, \mathbf{V}, \mathbf{H}, \mathbf{W} \geq 0$ ;
2 while Not convergent and iter $\leq I$ do
3     Update $\mathbf{H}(i,j) \leftarrow \mathbf{H}(i,j)\sqrt{\frac{[\mathbf{U}^T\mathbf{X}\mathbf{V}](i,j)}{[\mathbf{U}^T\mathbf{U}\mathbf{H}\mathbf{V}^T\mathbf{V}](i,j)}}$ ;
4     Update $\mathbf{U}(i,j) \leftarrow \mathbf{U}(i,j)\sqrt{\frac{[\mathbf{X}\mathbf{V}\mathbf{H}^T+\mathbf{U}\Gamma_U^-](i,j)}{[\mathbf{U}\mathbf{H}\mathbf{V}^T\mathbf{V}\mathbf{H}^T+\mathbf{U}\Gamma_U^+](i,j)}}$ ;
5     Update $\mathbf{V}(i,j) \leftarrow \mathbf{V}(i,j)\sqrt{\frac{[\mathbf{X}^T\mathbf{U}\mathbf{H}+\lambda\mathbf{Y}\mathbf{W}^T+\mathbf{V}\Gamma_V^-](i,j)}{[\mathbf{V}\mathbf{H}^T\mathbf{U}^T\mathbf{U}\mathbf{H}+\lambda\mathbf{V}\mathbf{W}\mathbf{W}^T+\mathbf{V}\Gamma_V^+](i,j)}}$ ;
6     Update $\mathbf{W}(i,j) \leftarrow \mathbf{W}(i,j)\sqrt{\frac{[\mathbf{V}^T\mathbf{Y}](i,j)}{[\mathbf{V}^T\mathbf{V}\mathbf{W}](i,j)}}$ ;
7     iter = iter + 1 ;
8 end
```

Algorithm 4: *Modeling Knowledge across Media*

which leads to the updating rule of **W**,

$$\mathbf{W}(i,j) \leftarrow \mathbf{W}(i,j)\sqrt{\frac{[\mathbf{V}^T\mathbf{Y}](i,j)}{[\mathbf{V}^T\mathbf{V}\mathbf{W}](i,j)}}. \tag{7.23}$$

I summarize the algorithm of optimizing Eq. (7.3) in Algorithm 4, where I is the number of maximum iterations. In line 1, I conduct initialization for the variables. From lines 2 to 8, the four variables are updated with the updating rules until convergence or until they reach the number of maximum iterations. The correctness and convergence of the updating rules can be proven with the standard auxiliary function approach [88].

7.4.2 Modeling Content Information

In this subsection, as shown in the lower right part of Figure 7.1, I introduce how to model content information of microblogging data in the proposed model.

To make use of the content information of microblogging messages, I introduce a graph Laplacian [19] in the proposed model. I construct a graph based on content information of the users. In the graph, each node represents a user and each edge represents the affinity between two users. The adjacency matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ of the graph is defined as

$$\mathbf{M}(u,v) = \begin{cases} 1 & \text{if } \; u \in \mathcal{N}(v) \; or \; v \in \mathcal{N}(u) \\ 0 & \text{otherwise} \end{cases} \tag{7.24}$$

where u and v are nodes, and $\mathcal{N}(u)$ represents the k-nearest neighbor of the user. Content similarity is adopted to obtain the k-nearest neighbor in this work. Since the aim is to model the mutual content similarity between two users, the adjacency matrix is symmetric.

The basic idea of of using the graph Laplacian to model the content information is that if two nodes are close in the graph, i.e., they posted similar messages, their identity labels should be close to each other. It can be mathematically formulated as minimizing the following loss function:

$$\mathcal{R} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \|\mathbf{V}_t(i,*) - \mathbf{V}_t(j,*)\|_2^2 \mathbf{M}(i,j). \tag{7.25}$$

This loss function will incur a penalty if two users have different predicted labels when they are close to each other in the graph. Let $\mathbf{D} \in \mathbb{R}^{n \times n}$ denote a diagonal matrix, and its diagonal element is the degree of a user in the adjacency matrix $\mathbf{M}$, i.e., $\mathbf{D}(i,i) = \sum_{j=1}^{n} \mathbf{M}(i,j)$.

Theorem 3 *The formulation in Eq. (7.25) is equivalent to the following objective function:*

$$\mathcal{R} = Tr(\mathbf{V}_t^T \mathcal{L} \mathbf{V}_t), \tag{7.26}$$

where the Laplacian matrix [19] $\mathcal{L}$ is defined as $\mathcal{L} = \mathbf{D} - \mathbf{M}$.

Proof. It is easy to verify that Eq. (7.25) can be rewritten as

$$\begin{aligned} \mathcal{R} = \; & \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{c} \mathbf{V}_t(i,k)\mathbf{M}(i,j)\mathbf{V}_t^T(i,k) - \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{c} \mathbf{V}_t(i,k)\mathbf{M}(i,j)\mathbf{V}_t^T(j,k) \\ = \; & Tr(\mathbf{V}_t^T(\mathbf{D} - \mathbf{M})\mathbf{V}_t) \\ = \; & Tr(\mathbf{V}_t^T \mathcal{L} \mathbf{V}_t), \end{aligned} \tag{7.27}$$

which completes the proof. □

```
Input: {$\mathbf{T}, \mathbf{U}, \alpha, \beta, I$ }
Output: $\mathbf{V}_t$
Construct matrices $\mathcal{L}$ in Eq. (7.26) ;
Initialize $\mathbf{U}_t = \mathbf{U}, \mathbf{V}, \mathbf{H} \geq 0$ ;
while Not convergent and iter $\leq I$ do
    Update $\mathbf{H}_t(i, j) \leftarrow \mathbf{H}_t(i, j) \sqrt{\frac{[\mathbf{U}_t^T \mathbf{X} \mathbf{V}_t](i,j)}{[\mathbf{U}_t^T \mathbf{U}_t \mathbf{H}_t \mathbf{V}_t^T \mathbf{V}_t](i,j)}}$ ;
    Update $\mathbf{U}_t(i, j) \leftarrow \mathbf{U}_t(i, j) \sqrt{\frac{[\mathbf{X} \mathbf{V}_t \mathbf{H}_t^T + \beta \mathbf{G}_U \mathbf{U} + \mathbf{U}_t \Gamma_U^-](i,j)}{[\mathbf{U}_t \mathbf{H}_t \mathbf{V}_t^T \mathbf{V}_t \mathbf{H}_t^T + \beta \mathbf{G}_U \mathbf{U}_t + \mathbf{U}_t \Gamma_U^+](i,j)}}$ ;
    Update $\mathbf{V}_t(i, j) \leftarrow \mathbf{V}_t(i, j) \sqrt{\frac{[\mathbf{X}^T \mathbf{U}_t \mathbf{H}_t + \alpha \mathbf{M} \mathbf{V}_t + \mathbf{V}_t \Gamma_V^-](i,j)}{[\mathbf{V}_t \mathbf{H}_t^T \mathbf{U}_t^T \mathbf{U}_t \mathbf{H}_t + \alpha \mathbf{D} \mathbf{V}_t + \mathbf{V}_t \Gamma_V^+](i,j)}}$ ;
    $iter = iter + 1$;
end
```

Algorithm 5: *Spammer Detection in Microblogging*

7.4.3 Spammer Detection Framework

As illustrated in Figure 7.1, I employ two types of information to formulate two kinds of constraints on the learned model. By integrating knowledge learned from other media and content information from microblogging, spammer detection can be performed by optimizing

$$\begin{aligned} \min_{\mathbf{U}_t, \mathbf{H}_t, \mathbf{V}_t \geq 0} \quad & \mathcal{J} = \|\mathbf{T} - \mathbf{U}_t \mathbf{H}_t \mathbf{V}_t^T\|_F^2 + \alpha Tr(\mathbf{V}_t^T \mathcal{L} \mathbf{V}_t) + \beta \|\mathbf{G}_U (\mathbf{U}_t - \mathbf{U})\|_F^2), \\ s.t. \quad & \mathbf{U}_t^T \mathbf{U}_t = \mathbf{I}, \;\; \mathbf{V}_t^T \mathbf{V}_t = \mathbf{I}, \end{aligned} \tag{7.28}$$

where the first term is to factorize the microblogging data into three variables, which are similar to the idea discussed in Section 7.4.1. The second term is to introduce content information and the third is to introduce knowledge learned from the other media. $\mathbf{U}$ is the lexicon learned from the other media by solving the problem in Eq. (7.3). $\mathbf{G}_U \in \{0, 1\}^{m \times m}$ is a diagonal indicator matrix to control the impact of the learned lexicon, i.e., $\mathbf{G}_U(i, i) = 1$ represents that the i-th word contains identity information, $\mathbf{G}_U(i, i) = 0$ otherwise.

Table 7.4: Spammer Detection Results on TweetH Dataset

	External Data I (50%)			*External Data II (100%)*		
	Precision	Recall	F_1-measure (gain)	Precision	Recall	F_1-measure (gain)
Least_Squares	0.823	0.834	0.828 (N.A.)	0.839	0.852	0.845 (N.A.)
Lasso	0.865	0.891	0.878 (+5.96%)	0.873	0.905	0.889 (+5.12%)
MFTr	0.866	0.899	0.882 (+6.49%)	0.887	0.918	0.902 (+6.72%)
MFSD	0.644	0.703	0.672 (-18.7%)	0.650	0.715	0.681 (-19.5%)
CSD	0.906	0.939	0.922 (+11.3%)	0.913	0.944	0.928 (+9.79%)

Table 7.5: Spammer Detection Results on TweetS Dataset

	External Data I (50%)			*External Data II (100%)*		
	Precision	Recall	F_1-measure (gain)	Precision	Recall	F_1-measure (gain)
Least_Squares	0.766	0.813	0.789 (N.A.)	0.793	0.820	0.806 (N.A.)
Lasso	0.801	0.849	0.824 (+4.50%)	0.814	0.848	0.831 (+3.02%)
MFTr	0.810	0.857	0.833 (+5.58%)	0.833	0.878	0.855 (+6.03%)
MFSD	0.621	0.69	0.654 (-17.1%)	0.642	0.681	0.661 (-18.0%)
CSD	0.832	0.875	0.853 (+8.13%)	0.848	0.919	0.882 (+9.40%)

This optimization problem is not convex with respect to the three parameters together. Following the optimization procedure to solve Eq. (7.3), I propose an algorithm to solve the problem in Eq. (7.28) and summarize it in Algorithm 5. In line 1, I construct the Laplacian matrix $\mathcal{L}$. In line 2, I initialize the variables. From lines 3 to 9, I keep updating the variables with the updating rules until convergence or until the number of maximum iterations is reached.

7.5 Evaluation

In this section, I empirically evaluate the proposed learning framework and the factors that could bring in effects to the framework. Through the experiments, the aim is to answer the following two questions:

- How effective is the proposed framework compared with other possible solutions of using external information across media in real-world spammer detection tasks?
- What impact do the other resources have on the performance of spammer detection in microblogging?

7.5.1 Experimental Settings

I follow a standard experiment setup used in spammer detection literature [110] to evaluate the effectiveness of the proposed framework for leveraging knowledge aCross media for Spammer Detection (*CSD*). In particular, I compare the proposed framework *CSD* with different baseline methods for spammer detection. To avoid bias, both TweetH and TweetS, introduced in Section 7.2.1, are used in the experiments. For email data, I consider each sender a user; For SMS and web data, I do not have user information and consider each message as sent from a distinct user. In the experiment, precision, recall and F$_1$-measure are used as the performance metrics.

To evaluate the general performance of the proposed framework, I use all of the three datasets from different media, i.e., Email, SMS and Web datasets. In the first set of experiments, to be discussed in Section 7.5.2, I simply combine them together and consider them as homogeneous data sources. In the second set of experiments, to be discussed in Section 7.5.3, I consider their individual impact on the performance of spammer detection. A standard procedure for data preprocessing is used in the experiments. The unigram model is employed to construct the feature space, tf-idf is used as the feature weight.

As we have discussed in Sections 7.4.1 and 7.4.3, three positive parameters are involved in the experiments, including λ in Eq. (7.3), and α and β in Eq. (7.28). λ is to control the effect of knowledge from other media to the learned lexicon, α is to control the contribution of Laplacian regularization, and β is to control the contribution of lexicon to the spammer detection model. Since all the parameters can be tuned via cross-validation with a set of validation data, in the experiment, I empirically set $\lambda = 0.1$, $\alpha = 0.1$ and $\beta = 0.1$ for general experiment purposes. The effects of the parameters on the learning model will be further discussed in Section 7.5.4.

7.5.2 Effectiveness of the Proposed Method

I compare the proposed method *CSD* with other methods for spammer detection, accordingly answer the first question asked above. The baseline methods are listed below.

- *Least_Squares*: One possible solution for the task is to consider it as a supervised learning problem. I simply train a classification model with the available external data and apply the learned model on microblogging data for spammer detection. The widely used classifier, *Least_Squares* [32], is used for comparison.

- *Lasso*: Sparse learning methods are effective for high-dimensional data. I further include *Lasso* [95] as the baseline method, which performs continuous shrinkage and automatic feature selection by adding l_1 norm regularization to the Least Squares.

- *MFTr*: Although I first present a quantitative linguistic variation analysis and provide a unified model for spammer detection across different media, domain adaption and transfer learning have received great attention in various applications [83]. I apply a widely used transfer learning method [66], which transfers the knowledge directly from labeled data in the source domain to the target domain for classification, to test its performance on spammer detection in the experiment.

- *MFSD*: I test the performance of the unsupervised learning method by employing the basic matrix factorization model *MFSD*. This is a variant of the proposed method without introducing any knowledge learned from external sources. As a common initialization for clustering methods, I randomly assign initial centroids and an initial class indicator matrix for *MFSD*.

Experimental results of the methods on the two datasets, TweetH and TweetS, are respectively reported in Table 7.4 and 7.5. To avoid bias brought by the sizes of the training data,[6] I conduct two sets of experiments with different numbers of training instances. In the experiments, "External Data I (50%)" means that I randomly chose 50% from the whole training data. "External Data II (100%)" means that I use all the data for training. Also, "gain" represents the percentage improvement of the methods in comparison with the first baseline method *Least_Squares*. In the experiment, each result denotes an average of 10 test runs. By comparing the spammer detection performance of different methods, I observe the following:

(1) From the results in the tables, I can observe that the proposed method *CSD* consistently outperforms other baseline methods on both datasets with different sizes of training data. The proposed method achieves better results than the state-of-the-art method *MFTr* on both datasets. I apply two-sample one-tail t-tests to compare *CSD* to the four baseline methods. The experiment results demonstrate that the proposed model performs significantly better (with significance level $\alpha = 0.01$) than the four methods.

(2) The performance of the proposed method *CSD* is better than the first three baselines, which are based on different strategies of using resources from the other media. This demonstrates the excellent use of cross-media knowledge in the proposed framework for spammer detection.

[6] Similar to the definitions in machine learning literature, training data here refers to the labeled data from the external sources, and testing data represents the unlabeled microblogging data.

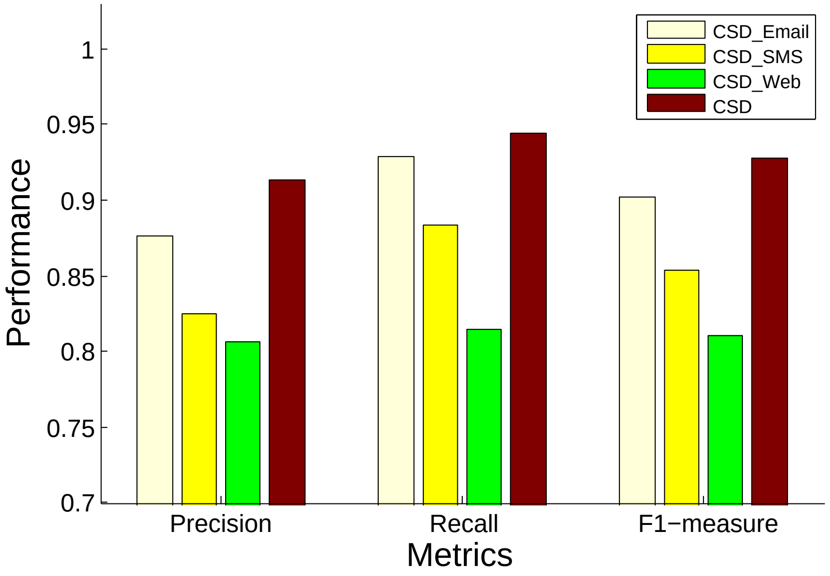


Figure 7.2: Results on TweetH Dataset

(3) Among the baseline methods, *MFTr* achieves the best results. It demonstrates that the knowledge transferred from other media help the task of spammer detection in microblogging. *Lasso* performs better than *Least_Squares*. This shows that, for high-dimensional textual data from email, SMS and web, feature selection is necessary for a supervised learning method for this task I study.

(4) The method *MFSD* achieves the worst performance among all the baseline methods. It shows that learning based on microblogging data itself can not discriminant well between spammers and legitimate users. It further demonstrates that the knowledge learned from external sources is helpful to build an effective model to tackle the problem.

In summary, with the effective use of data from the other media, the proposed framework outperforms the baseline methods in spammer detection. Next, I investigate the effects of different resources on the spammer detection task.

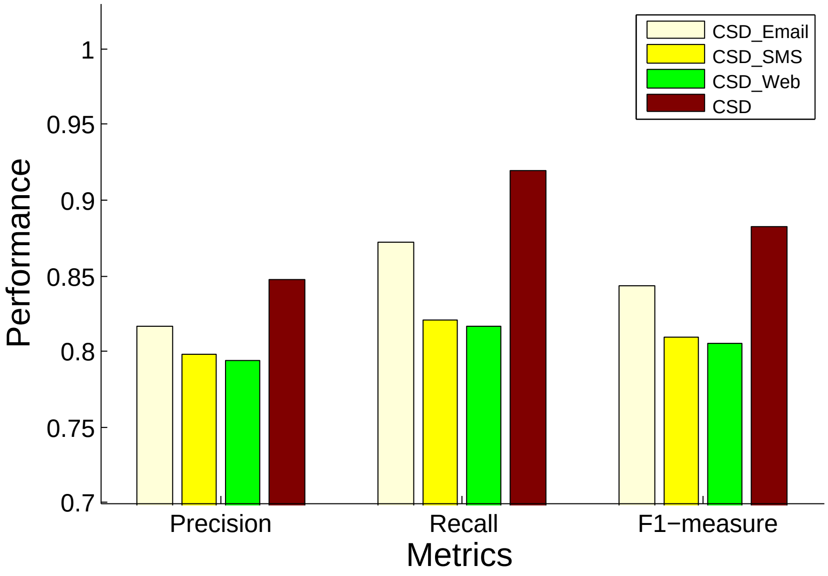


Figure 7.3: Results on TweetS Dataset

Table 7.6: Learning from Different Media for Spammer Detection in Microblogging

	Email	SMS	Web	TweetH (loss)	TweetS (loss)
Default	1	1	1	0.928 (N.A.)	0.882 (N.A.)
Knock Out One Term	0	1	1	0.881 (-5.09%)	0.843 (-4.43%)
	1	0	1	0.911 (-1.86%)	0.856 (-2.96%)
	1	1	0	0.923 (-0.57%)	0.860 (-2.50%)

7.5.3 Effectiveness of External Information Sources

In this subsection, I study the effects of the external information from the other media on the proposed framework, accordingly answering the second question asked in the beginning of Section 7.5.

I first evaluate the performance of the proposed framework with data from only one of the three media. In particular, I learn a lexicon based on one of the three types of media, i.e., email, SMS and web, and perform spammer detection on the microblogging datasets. I do not have legitimate web pages in the original Web dataset. To build a classifier *CSD_Web*, following the data construction procedure proposed in [46], I randomly sample 20,100 web snippets with BingAPI as legitimate data. The experimental results of the methods on the two microblogging datasets are plotted in Figures 7.2 and 7.3, respectively. In the figures, the first three bars represent the performance of the baselines with one type of external information. The last is the method with all three types of external information. From the figures, I observe the following:

(1) With the integration of all three types of external information, *CSD* consistently achieves better performance than the three baselines with only one type of information. It demonstrates that the proposed method uses beneficial information to perform effective spammer detection.

(2) Among the three baseline methods, *CSD_Email* and *CSD_SMS* achieve better performance than *CSD_Web*. It shows that, as external resources, email and SMS data are more suitable to be used for the spammer detection in microblogging than the web data. This result is consistent with the linguistic variation analysis in Section 7.2.

To explore the effects of different media sources on the performance of spammer detection, I employ a "knockout" technique in the experiment. Knockout has been widely used in many fields, e.g., gene function analysis, to test the performance variance brought by one component when it is made inoperative in the framework [27]. I conduct the experiments by knocking out one type of the external information from the proposed framework. The results are summarized in Table 7.6. In the table, "loss" represents the performance decrease of the methods as compared to the setting "Default" which is learned based on data from all three media sources. The three columns in the middle are experimental settings, in which "0" means this resource is knocked out. The last two columns are the F_1-measure results under different experimental settings. From the table, we observe the following:

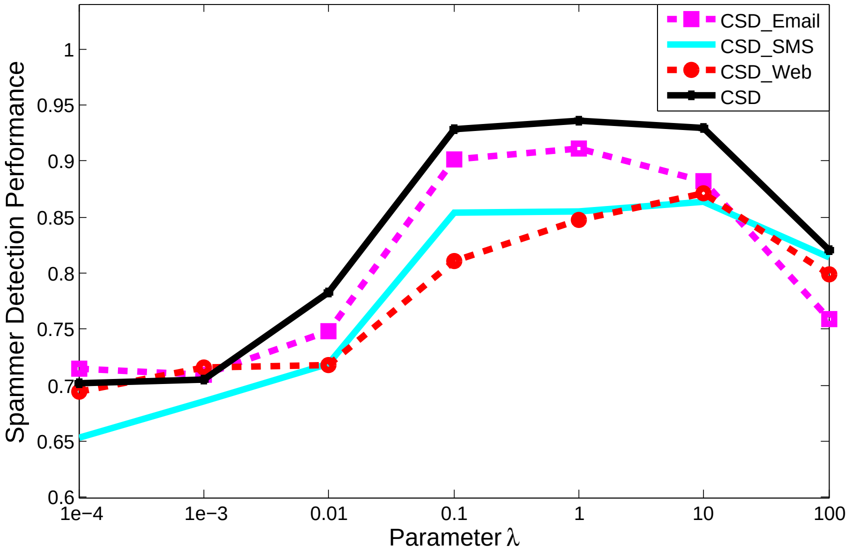


Figure 7.4: Performance with Different λ Settings

(1) By knocking out one of the external sources, performance of the proposed framework decreases. This suggests that all the three types of external information are useful for spammer detection in microblogging.

(2) Knocking out email from the resources incurs the most performance decrease among all the experimental settings. This demonstrates that email is the most effective source among the three types of information. This finding is consistent with the discussion above.

From the discussion above, it suggests that the use of data from the other media shows the effectiveness in spammer detection task. The superior performance of the proposed method *CSD* validates its excellent use of knowledge from the other media.

7.5.4 Discussion

Three positive parameters, i.e., λ, α and β, are involved in the proposed framework. I first examine the effects brought by λ, which is to control the contribution of knowledge from other media to the learned lexicon. In previous subsections, for general experimental

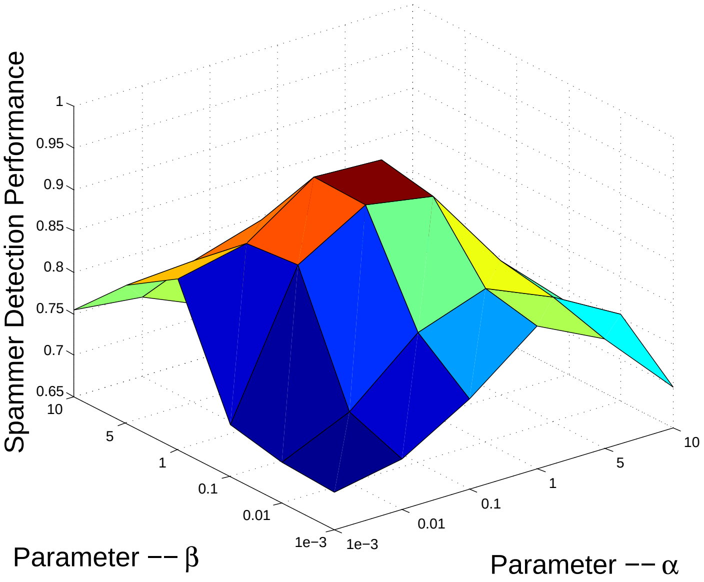


Figure 7.5: Impact of Content Information (α) and External Information (β)

purposes, I empirically set $\lambda = 0.1$. I now conduct experiments to compare the spammer detection performance of the four methods introduced in Section 7.5.3 with different settings of λ. The experiment results on the TweetH dataset are plotted in Figure 7.4. From the figure, I observe the following: (1) The general trends of the four methods are similar with the variation of different parameter settings. They achieve relatively good performance when setting λ in the range of [0.1, 10]. (2) In most cases, performance of the proposed *CSD* is better than the other three methods. It demonstrates that the combination of the three resources improve the spammer detection performance.

I further examine the effects of the parameters α and β discussed in Eq. (7.28) on the proposed framework. α is to control the contribution of content information and β is to control the effects of external information from the other media. To understand the effects brought by the parameters, I compare the spammer detection performance of the proposed *CSD* on the Twitter datasets with different parameter settings. The results on the TweetH dataset are plotted in Figure 7.5. From the figure, we observe that the proposed method

CSD performs well when $\alpha \in [0.1, 5]$ and $\beta \in [0.1, 1]$. Generally, the performance of *CSD* is not quite sensitive to the parameters. The proposed framework can perform well when choosing parameter settings in a reasonable range. Similar results have been observed for the two sets of experiments on the TweetS dataset.

7.6 Summary

Texts in microblogging are short, noisy, and labeling processing is time-consuming and labor-intensive, which presents great challenges for spammer detection. In this chapter, I first conduct a quantitative analysis to study how noisy the microblogging texts are by comparing them with spam messages from other media. The results suggest that microblogging data is not significantly different from data from the other media. Based on the observations, a matrix factorization model is employed to learn lexicon information from external spam resources. By incorporating external information from other media and content information from microblogging, I propose a novel framework for spammer detection. The experimental results demonstrate the effectiveness of the proposed model as well as the roles of different types of information in spammer detection.

Different from the discussion in last chapter, I study the problem of tackling labeling bottleneck from another aspect by learning knowledge from heterogeneous data domains in this chapter. Thus, I provide two options for tackling this problem. If we need label data from the same domain, to efficiently obtain informative and representative labeled data, active learning can be used; If it is not possible, we can just study the impassibility of using existing resources by cross-media learning. With these two ways, we can mitigate the widely existed problem to some extent.

Chapter 8

CONCLUSION AND FUTURE WORK

This chapter concludes the dissertation by summarizing the contributions of the work and highlighting the future directions.

8.1 Conclusion

With the growing availability of social media services, social spammer detection is becoming an important problem and attracts a lot of attention from academia and industry. Social Spammers send out unwanted spam content appearing on social networks and any website with user-generated content to targeted users, often corroborating to boost their social influence, legitimacy, credibility. Successful detecting spammers in social media is important to improve the quality of user experience, and to promote the healthy use and development of a social networking system. Social spammers are observed to *consist of heterogeneous information, contain contextual information, evolve very fast, and lack label data*. By tacking the data with distinct characteristics, our contributions can be summarized from two aspects: First, social spammer detection is a novel and practical problem. At the same time, it is challenging due to the characteristics of the data. I firstly present a systematical study to formalize the challenges of this novel problem. Second, by formally understanding the problem, I am able to propose novel and effective computational models to tackle the challenges and achieve good performance. In conclusion, in the dissertation, I investigate the problem from different perspectives to characterize and detect social spammers.

I investigated how to seamlessly integrate the heterogeneous information sources, network and content information of social media users, to perform effective social spammer detection. Spammer detection has been studied in various online social networking (OSN) platforms. One effective way to perform spammer detection is to utilize the social network information. The basic idea is that spammers cannot establish an arbitrarily large number of social trust relations with normal users. However, due to the unidirectional user binding in many online social networking systems and the reflexive reciprocity, it is possible for spammers to imitate normal users by quickly accumulating reputation on the social network. Meanwhile, social media provide additional content information other than the social networks. The proposed framework models both types of information in a unified way. Also, I present an efficient algorithm to solve the proposed non-smooth convex optimization problem. Experiments on a real Twitter dataset show that the proposed framework can effectively integrate both kinds of information to outperform the state-of-the-art methods.

Motivated by existing findings from psychology and social sciences, I study the problem of utilizing sentiment information for effective social spammer detection. Microexpressions have been used for detecting deception. Microexpression is usually observed when a person is consciously trying to conceal all signs of how he or she is feeling. To study this problem, I formulated hypothesis testing experiments to validate the assumption that spamming is associated with sentiment information. Further, I proposed a novel framework to incorporate sentiment information for social spammer detection. Through extensive experiments, it shows that the incorporation of sentiment information brings in positive effect to the performance of social spammer detection.

Given the rapidly evolving nature of social spammers, I proposed a framework that efficiently reflects the effect of newly emerging data in social spammer detection. Social spammers show dynamic content patterns and many existing social media sites have become a target of link farming. Spammers change their spamming strategies and pretend to be normal or even influential users to game the system. Building a new model to capture newly emerging content-based and network-based patterns of social spammers is inefficient. To handle those fast evolving social spammers, I investigate its online updating scheme and provide an efficient way for social spammer detection. Experimental results show that our proposed method is efficient in the model learning.

Many supervised learning methods suffer from the lack of label information in real-world applications. It presents great challenges for social spammer detection when there is no sufficient label data. In this dissertation, I proposed two methods to tackle the "labeling bottleneck". First, I propose the global selection and local selection strategies to find the most informative and representative data instances to query for label. This is a novel active learning framework to handle the networked texts in social media. Second, motivated by publicly available resources from other well-studied platforms, I proposed to learn cross-media knowledge for social spammer detection. I conducted lexical analysis to compare how different the resources are from the perspective of linguistic analysis. Based on the observations, a matrix factorization based framework is employed to learn knowledge from cross-media resources for spammer detection in microblogging. Experimental results on real-world datasets showed the effectiveness of the proposed method in tackling the labeling bottleneck.

Through this systematic study, on one hand, I focus on threats to these systems and design methods to mitigate negative behaviors; on the other, I look for positive opportunities to mine and analyze these systems for developing next generation algorithms and architectures that can empower decision makers.

8.2 Future Work

This work can be extended along these future directions.

(1) Spam-resilient Social Recommendation.

Recommender systems play an important role in helping online users find relevant information by suggesting information of potential interest to them. Due to the potential value of social relations in recommender systems, social recommendation [91] has attracted increasing attention in recent years. The rise of social spamming is jeopardizing trustworthy social recommendation. Given the exclusive reliance of existing social recommendation techniques on account activities, the spammers could be coordinated to create many seemingly legitimate account activities to inflate their influence in social media. Since influence scores are normally measured in a relative sense, the spammers could also easily deflate the influence scores of legitimate users. The failure to identify truly influential and trustable users due to spamming would inevitably hinder the use of social systems for effective information dissemination and sharing. Thus I plan to investigate a challenging interdisciplinary research plan on designing, prototyping, and evaluating a spam-resilient social recommendation framework.

(2) Characterizing Misinformation during Mass Emergency.

As an information dissemination platform social media has been used with varying success in several recent crises and mass emergency situations, as evidenced by many recent events like Hurricane Sandy and the Occupy Wall Street movement. The continued usage of social media as a platform to submit crisis related information motivates us to use relevant information as sensors of the real world. It is also observed that during 2010 earthquake in Chile and 2012 Hurricane Sandy many rumors were posted and spread on Twitter when the official sources are scarce. The misinformation leads to increasing the sense of chaos and insecurity in the local population during mass emergency. I propose to study the characteristics of misinformation during these events from two sides, the content information and social network information. Different from other spam messages, the misinformation should be strongly correlated with the associated events, thus its content information should contain distinguishable patterns. Understanding why people create and spread misinformation is also interesting aspect of the project.

(3) Data Analytics for Heterogeneous Information Sources.

Features from different sources could be strong indicators for various tasks in social media data analysis. For example, time zone, location, url, hashtag and length of the posts could be used as features for spammer detection in social media. However, how to effectively build a learning model for heterogeneous features is still an open problem in many applications. This task consists of two components, i.e. feature engineering and learning with heterogeneous information sources. I propose to profile [45] a social media user with features from three categories. First is the user information, e.g., registration age, whether this is a verified user, whether he posts description in the profile and gender information. Second is the posting behavior, e.g., when and where he usually posts, how frequent he posts, whether he likes retweet and whether he likes "@" his friends. Third is the topic information, e.g., hatags and sentiment of the posts. How to integrate the features in an unified model is still an open problem. Our preliminary study [92] proposed to select features from heterogeneous feature spaces by exploiting relations among the sources. I will study along this direction to investigate how to exploit link information in social media for heterogeneous data analysis.

REFERENCES

[1] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami. Contributions to the study of sms spam filtering: new collection and results. In *Proceedings of DocEng*, 2011.

[2] M. Balcan, A. Broder, and T. Zhang. Margin based active learning. *Learning Theory*, pages 35–50, 2007.

[3] T. Baldwin, P. Cook, M. Lui, A. MacKinlay, and L. Wang. How noisy social media text, how diffrnt social media sources? In *Proceedings of IJCNLP*, 2013.

[4] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In *Proceedings of CEAS*, 2010.

[5] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of WWW*, 2009.

[6] M. Bilgic, L. Mihalkova, and L. Getoor. Active learning for networked data. In *ICML*, 2010.

[7] E. Blanzieri and A. Bryl. A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review*, 29(1):63–92, 2008.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the JMLR*, 3:993–1022, 2003.

[9] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2011.

[10] J. Bollen, A. Pepe, and H. Mao. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *CoRR, abs/0911.1583*, 2009.

[11] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. The socialbot network: when bots socialize for fame and money. In *ACSAC*, 2011.

[12] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[13] P. Boykin and V. Roychowdhury. Leveraging social networks to fight spam. *Computer*, 38(4):61–68, 2005.

[14] H. M. Breland. Word frequency and word difficulty: A comparison of counts in four corpora. *PSS*, 1996.

[15] G. Brown, T. Howe, M. Ihbe, A. Prakash, and K. Borders. Social networks and context-aware spam. In *Proceedings of CSCW*, 2008.

[16] S. S. Bucak and B. Gunsel. Incremental subspace learning via non-negative matrix factorization. *Pattern Recognition*, 42(5):788–797, 2009.

[17] C. Chang and C. Lin. Libsvm: a library for support vector machines. *TIST*, 2011.

[18] Z. Chen, K. A. Wilson, Y. Jin, W. Hendrix, and N. F. Samatova. Detecting and tracking community dynamics in evolutionary networks. In *ICDMW*, pages 318–327, 2010.

[19] F. Chung. *Spectral graph theory*. Number 92. Amer Mathematical Society, 1997.

[20] F. Chung. Laplacians and the cheeger inequality for directed graphs. *Annals of Combinatorics*, 9(1):1–19, 2005.

[21] D. Cohn, Z. Ghahramani, and M. Jordan. Active learning with statistical models. *Arxiv*, 1996.

[22] I. Dagan and S. Engelson. Committee-based sampling for training probabilistic classifiers. In *ICML*, 1995.

[23] G. Danezis and P. Mittal. Sybilinfer: Detecting sybil nodes using social networks. 2009.

[24] L. Danon, J. Duch, A. Arenas, and A. Daz-guilera. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005.

[25] S. Dasgupta and D. Hsu. Hierarchical sampling for active learning. In *ICML*, 2008.

[26] C. Ding, T. Li, and M. Jordan. Convex and semi-nonnegative matrix factorizations. *TPAMI*, 2010.

[27] T. Egener, J. Granado, and M. Guitton. High frequency of phenotypic deviations in physcomitrella patens plants transformed with a gene-disruption library. *BMC Plant Biology*, 2:6, 2002.

[28] J. Eisenstein. What to do about bad language on the internet. In *Proceedings of NAACL-HLT*, 2013.

[29] P. Ekman. *Telling lies: Clues to deceit in the marketplace, politics, and marriage*. WW Norton & Company, 2009.

[30] P. Ekman and W. V. Friesen. *Unmasking the face: A guide to recognizing emotions from facial clues*. Ishk, 2003.

[31] Y. Freund, H. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2):133–168, 1997.

[32] J. Friedman, T. Hastie, and R. Tibshirani. The elements of statistical learning, 2008.

[33] L. Ghanoui, G. Li, V. Duong, V. Pham, A. Srivastava, and K. Bhaduri. Sparse machine learning methods for understanding large text corpora. In *Proceedings of CIDU*, 2011.

[34] S. Ghosh, B. Viswanath, F. Kooti, N. Sharma, G. Korlam, F. Benevenuto, N. Ganguly, and K. Gummadi. Understanding and combating link farming in the twitter social network. In *Proceedings of WWW*, 2012.

[35] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *Technical Report, Stanford*, 2009.

[36] J. M. Gómez Hidalgo, G. C. Bringas, E. P. Sánz, and F. C. García. Content based sms spam filtering. In *Proceedings of DocEng*, 2006.

[37] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @ spam: the underground on 140 characters or less. In *Proceedings of CCS*, 2010.

[38] Q. Gu and J. Han. Towards feature selection in network. In *Proceedings of CIKM*, 2011.

[39] Q. Gu, J. Zhou, and C. H. Ding. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *SDM*, pages 199–210, 2010.

[40] E. A. Haggard and K. S. Isaacs. Micro-momentary facial expressions as indicators of ego mechanisms in psychotherapy., 1966.

[41] M. A. Halliday and C. M. Matthiessen. An introduction to functional grammar. 2004.

[42] M. Hart. *Project gutenberg*. Project Gutenberg, 1971.

[43] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of SIGIR*, 1999.

[44] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.

[45] X. Hu and H. Liu. Mining and profiling in social media. In *International Encyclopedia of Digital Communication & Society*. Springer, 2014.

[46] X. Hu, N. Sun, C. Zhang, and T.-S. Chua. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *Proceedings of CIKM*, 2009.

[47] X. Hu, J. Tang, H. Gao, and H. Liu. Unsupervised sentiment analysis with emotional signals. In *Proceedings of the 22nd international conference on World Wide Web*, WWW'13. ACM, 2013.

[48] X. Hu, J. Tang, and H. Liu. Leveraging knowledge across media for spammer detection in microblogging. In *SIGIR*, 2014.

[49] X. Hu, J. Tang, Y. Zhang, and H. Liu. Social spammer detection in microblogging. In *IJCAI*, 2013.

[50] X. Hu, L. Tang, and H. Liu. Enhancing accessibility of microblogging messages using semantic knowledge. In *Proceedings of CIKM*, 2011.

[51] X. Hu, L. Tang, J. Tang, and H. Liu. Exploiting social relations for sentiment analysis in microblogging. In *WSDM*, 2013.

[52] Y. Hu, K. Talamadupula, and S. Kambhampati. Dude, srsly?: The surprisingly formal nature of twitters language. *Proceedings of ICWSM*, 2013.

[53] J. Huang, K. M. Thornton, and E. N. Efthimiadis. Conversational tagging in twitter. In *Hypertext*, 2010.

[54] S. Huang, R. Jin, and Z. Zhou. Active learning by querying informative and representative examples. *NIPS*, 2010.

[55] M. Ji and J. Han. A variance minimization criterion to active learning on graphs.

[56] S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *Proceedings of ICML*, 2009.

[57] N. Jindal and B. Liu. Opinion spam and analysis. In *Proceedings of WSDM*, 2008.

[58] X. Kong, W. Fan, and P. Yu. Dual active feature and sample selection for graph classification. In *Proceedings of SIGKDD*, 2011.

[59] S. Kumar, F. Morstatter, and H. Liu. *Twitter data analytics*. Springer, 2014.

[60] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of WWW*, 2010.

[61] C. Lawson and R. Hanson. *Solving least squares problems*, volume 15. SIAM, 1995.

[62] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, pages 788–791, 1999.

[63] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: social honeypots + machine learning. In *Proceedings of SIGIR*, 2010.

[64] K. Lee, B. D. Eoff, and J. Caverlee. Seven months with the devils: A long-term study of content polluters on twitter. In *Proceedings of ICWSM*, 2011.

[65] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *SIGIR*, 1994.

[66] T. Li, V. Sindhwani, C. Ding, and Y. Zhang. Knowledge transformation for cross-domain sentiment classification. In *SIGIR*, 2009.

[67] B. Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 2012.

[68] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient l 2, 1-norm minimization. In *Proceedings of UAI*, 2009.

[69] Y. Lu, M. Castellanos, U. Dayal, and C. Zhai. Automatic construction of a context-aware sentiment lexicon: an optimization approach. In *Proceedings of WWW*, 2011.

[70] O. Madani, H. H. Bui, and E. Yeh. Efficient online learning and prediction of users' desktop actions. In *IJCAI*, 2009.

[71] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of ICML*, 2009.

[72] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *The JMLR*, 2010.

[73] T. Marinis. Psycholinguistic techniques in second language acquisition research. *Second Language Research*, 19(2):144–161, 2003.

[74] D. Matsumoto, H. S. Hwang, L. Skinner, and M. Frank. Evaluating truthfulness and detecting deception. *FBI Law Enforcement Bulletin, June*, pages 1–11, 2011.

[75] V. Metsis, I. Androutsopoulos, and G. Paliouras. Spam filtering with naive bayes-which naive bayes? In *Proceedings of CEAS*, 2006.

[76] Y. Nesterov and I. Nesterov. *Introductory lectures on convex optimization: A basic course*. 2004.

[77] M. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[78] M. Newman. *Networks: an introduction*. Oxford University Press, Inc., 2010.

[79] H. Nguyen and A. Smeulders. Active learning using pre-clustering. In *ICML*, 2004.

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

[81] D. O'Callaghan, M. Harrigan, J. Carthy, and P. Cunningham. Network analysis of recurring youtube spam campaigns. In *ICWSM*, 2012.

[82] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. *Technical Report, Stanford*, 1999.

[83] S. J. Pan and Q. Yang. A survey on transfer learning. *TKDE*, pages 1345–1359, 2010.

[84] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of ACL and EMNLP*, 2002.

[85] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, A. Flammini, and F. Menczer. Detecting and tracking political abuse in social media. In *Proceedings of ICWSM*, 2011.

[86] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, S. Patil, A. Flammini, and F. Menczer. Truthy: mapping the spread of astroturf in microblog streams. In *Proceedings of WWW*, 2011.

[87] B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 2010.

[88] D. Seung and L. Lee. Algorithms for non-negative matrix factorization. *NIPS*, 2001.

[89] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou, and P. Li. User-level sentiment analysis incorporating social networks. In *Proceedings of KDD*, 2011.

[90] J. Tang, H. Gao, X. Hu, and H. Liu. Exploiting homophily effect for trust prediction. In *Proceedings of WSDM*, 2013.

[91] J. Tang, X. Hu, H. Gao, and H. Liu. Exploiting local and global social context for recommendation. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence (IJCAI)*, 2013.

[92] J. Tang, X. Hu, H. Gao, and H. Liu. Unsupervised feature selection for multi-view data in social media. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 2013.

[93] L. Tang and H. Liu. Relational learning via latent social dimensions. In *Proceedings of KDD*, 2009.

[94] K. Thomas, C. Grier, D. Song, and V. Paxson. Suspended accounts in retrospect: An analysis of twitter spam. In *Proceedings of IMC*, 2011.

[95] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[96] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2002.

[97] D. Wang, R. Vipperla, N. Evans, and T. F. Zheng. Online non-negative convolutive pattern learning for speech signals. 2013.

[98] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang. Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In *Proceedings of CIKM*, 2011.

[99] R. Wardhaugh. *An introduction to sociolinguistics*, volume 28. Wiley. com, 2011.

[100] S. Webb, J. Caverlee, and C. Pu. Introducing the webb spam corpus: Using email spam to identify web spam automatically. In *CEAS*, 2006.

[101] S. Webb, J. Caverlee, and C. Pu. Social honeypots: Making friends with a spammer near you. In *Proceedings of CEAS*, 2008.

[102] J. Weng, E. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of WSDM*, 2010.

[103] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang. Representative sampling for text classification using support vector machines. *Information Retrieval*, 2003.

[104] H. Yang, Z. Xu, I. King, and M. R. Lyu. Online learning for group lasso. In *Proceedings of ICML*, 2010.

[105] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Zhao, and Y. Dai. Uncovering social network sybils in the wild. In *Proceedings of IMC*, 2011.

[106] S. J. Yates. Oral and written linguistic aspects of computer conferencing. *Pragmatics and beyond New Series*, 1996.

[107] K. Yu, S. Yu, and V. Tresp. Multi-label informed latent semantic indexing. In *Proceedings of SIGIR*, 2005.

[108] R. Zafarani, M. A. Abbasi, and H. Liu. *Social media mining: an introduction*. Cambridge University Press, 2014.

[109] D. Zhou, J. Huang, and B. Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *Proceedings of ICML*, 2005.

[110] Y. Zhu, X. Wang, E. Zhong, N. Liu, H. Li, and Q. Yang. Discovering spammers in social networks. In *AAAI*, 2012.

[111] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

BIOGRAPHICAL SKETCH

Xia Hu is a Ph.D. candidate of Computer Science and Engineering at Arizona State University. He received the BEng and MEng degrees in Computer Science and Engineering at Beihang University. His research interests are in data mining, social network analysis, machine learning, etc. As a result of his research work, he has published research papers in several major academic venues, including WWW, SIGIR, KDD, WSDM, IJCAI, AAAI, CIKM, SDM, etc. One of his papers was selected in the Best Paper Shortlist in WSDM'13. He is the recipient of the 2014 ASUs Presidents Award for Innovation, and Faculty Emeriti Fellowship. He has served on program committees for several major conferences such as IJCAI, SDM and ICWSM, and reviewed for multiple journals, including IEEE TKDE, ACM TOIS and Neurocomputing. His research attracts wide range of external government and industry sponsors, including NSF, ONR, AFOSR, Yahoo!, and Microsoft. Updated information can be found at `http://www.public.asu.edu/~xiahu`.