Misinformation Detection in Social Media

by

Liang Wu

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

Approved March 2019 by the
Graduate Supervisory Committee:

Huan Liu, Chair
Hanghang Tong
Adam Doupé
Brian D. Davison

ARIZONA STATE UNIVERSITY

March 2019

ABSTRACT

The pervasive use of social media gives it a crucial role in helping the public perceive reliable information. Meanwhile, the openness and timeliness of social networking sites also allow for the rapid creation and dissemination of misinformation. It becomes increasingly difficult for online users to find accurate and trustworthy information. As witnessed in recent incidents of misinformation, it escalates quickly and can impact social media users with undesirable consequences and wreak havoc instantaneously. Different from some existing research in psychology and social sciences about misinformation, social media platforms pose unprecedented challenges for misinformation detection. First, intentional spreaders of misinformation will actively disguise themselves. Second, content of misinformation may be manipulated to avoid being detected, while abundant contextual information may play a vital role in detecting it. Third, not only accuracy, earliness of a detection method is also important in containing misinformation from being viral. Fourth, social media platforms have been used as a fundamental data source for various disciplines, and these research may have been conducted in the presence of misinformation. To tackle the challenges, we focus on developing machine learning algorithms that are robust to adversarial manipulation and data scarcity.

The main objective of this dissertation is to provide a systematic study of misinformation detection in social media. To tackle the challenges of adversarial attacks, I propose adaptive detection algorithms to deal with the active manipulations of misinformation spreaders via content and networks. To facilitate content-based approaches, I analyze the contextual data of misinformation and propose to incorporate the specific contextual patterns of misinformation into a principled detection framework. Considering its rapidly growing nature, I study how misinformation can be detected at an early stage. In particular, I focus on the challenge of data scarcity and propose a novel framework to enable historical data to be utilized for emerging incidents that are seemingly irrelevant. With misinformation being viral, applications that rely on social media data face the challenge of corrupted data. To this end, I present robust statistical relational learning and personalization algorithms to minimize the negative effect of misinformation.

To my parents and wife for their love and support

Acknowledgements

I would like to thank my advisor, Huan Liu, for his continuous guidance and support during my PhD study. I received unreserved help on research guidance and for the achievement of personal goals. I would like to thank my dissertation committee members, Brian D. Davison, Hanghang Tong and Adam Doupé, for their valuable interactions and feedback.

I have been fortunate to work with many colleagues. I want to thank Xia Hu, Jundong Li, Fred Morstatter, Liang Du, Liangjie Hong, Mihajlo Grbovic, Justin Sampson, Kathleen M. Carley, Diane Hu, Harsh Dani, Kewei Cheng, Tahora H. Nazer, Sicong Kuang, and Giovanni Luca Ciampaglia.

Members of our Data Mining and Machine Learning Lab inspired me a lot through discussions, group meetings, and project collaborations. I would like to thank Ali Abbasi, Huiji Gao, Pritam Gundecha, Isaac Jones, Shamanth Kumar, Suhas Ranganath, Jiliang Tang, Robert Trevino, Suhang Wang, Reza Zafarani, Philippe Christophe Faucon, Yunzhong Liu, Ghazaleh Beigi, Kai Shu, Lu Cheng, Nur Shazwani Kamrudin, Ruocheng Guo, Kaize Ding, Raha Moraffah, Bing Hu, Matthew Davis, Alex Nou, and Daniel Howe, for their valuable suggestions and discussions.

I had the privilege of mentoring some terrific undergraduate and master students. Through The School of Computing, Informatics and Decision Systems Engineerings capstone program, I worked with Corey Mcneish, Christina Wilmot, Raquel Lippincott, Spencer Graf, Matthew Gross, and Philip Terzic. Through Barrett, The Honors Colleges Undergraduate Thesis program I had the opportunity to work with Lilian Ngweta. I have had the privilege of supervising master students who helped with our projects: Shobhit Sharma, Ashutosh Bhadke, Kunal Bansal, and Venkatesh Magham. I am grateful to all of the students for their assistance in the lab.

United States would not have been so enjoyable to settle down without the support of my dear friends. I would especially like to thank Xing Liang, Jundong Li, Fred Morstatter, Rio Cavendish and all of my other friends for their support.

Finally, to my parents for their love and support through my many years of graduate study. To my wife for always supporting my dreams, ideas and endeavors - none of my accomplishments are mine alone. To Frankie, our orange tabby friend, for adopting me and making our apartment a home.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

A rapid increase of social networking services in recent years has revolutionized the way people communicate and seek information. The openness of social media makes social networking applications, such as Facebook[1] and Twitter[2], a popular platform for the communication of trending and time-sensitive content. A recent study from Pew Research finds that 62% of adults get their news from social media in United States, with 29% among them doing so very often[3]. Therefore, it is of paramount importance to keep misinformation from being viral in social media.

However, the openness and increasing popularity of social networking platforms also make them an ideal target for misinformation dissemination. There are several related terms similar to misinformation. Rather than the concepts that are relatively easy to distinguish, such as spam and rumor, the most related term is disinformation, which specially refers to the intentionally spread incidents. In this dissertation, we refer to misinformation as an umbrella term to include all false or inaccurate information that is spread in social media. We choose to do so since on a platform where any user can publish anything, it is particularly difficult for researchers, or even administrators of social network companies, to determine whether a piece of misinformation is deliberately created. The concepts that are covered include disinformation, spam, rumor, fake news, and all of them share a characteristic that the inaccurate messages can causes distress and various kinds of destructive effects through social media.

[1] https://www.facebook.com/

[2] https://www.twitter.com/

[3] http://www.journalism.org/2016/05/26/news-use-across-social-media-platforms-2016/

There have been examples of widespread misinformation in social media during the 2016 Presidential Election in the US. An example of misinformation facilitating unnecessary fears through social media. One such example is **PizzaGate**, a conspiracy theory about a pizzeria being a nest of child-trafficking. It started breaking out simultaneously on multiple social media sites including Facebook, Twitter and Reddit[4]. After being promoted by radios and podcasts[5], the tense situation finally motivated someone to fire a rifle inside the restaurant[6]. PizzaGate even circulates for a while after the gunfire and being debunked. Though misinformation has been studied in psychology and social sciences, characteristics of social networking platforms, together with the adversarial attacks of misinformation spreaders present novel challenges for the task of misinformation detection in social media.

First, in the process of information propagation, misinformation spreaders would exploit vulnerabilities of social networking platforms to avoid being identified. Unlike traditional classification tasks of social media accounts, misinformation spreaders cannot be detected through directly modeling their content information or network topology. For example, since content on social networking websites is mostly accessible, a misinformation spreader can copy a great portion of content from legitimate users to disguise the malicious activities. In addition, since many users carelessly follow back when being followed on a social networking website, misinformation spreaders can gain friendship with legitimate users, which makes them difficult to distinguish in the networks. Last, label information is far less than sufficient for the task. For example,

[4] https://www.nytimes.com/2016/11/21/technology/fact-check-this-pizzeria-is-not-a-child-trafficking-site.html

[5] https://www.nytimes.com/2017/03/25/business/alex-jones-pizzagate-apology-comet-ping-pong.html

[6] https://www.nytimes.com/2016/12/05/us/pizzagate-comet-ping-pong-edgar-maddison-welch.html

the manipulation of content and network can boil down to a binomial classification problem given label information. However, considering the scale of social networking sites, it is impractical to collect labels for individual posts or links.

Second, distinct features of misinformation in social media make it difficult to directly apply classic content-based methods, while contextual information, which is abundant on social networking platforms, may provide us more effective features to characterize misinformation. For example, content of misinformation can be manipulated to be very similar to the content of true news and legitimate information (Piper, 2001). On the other hand, similar messages usually leads to similar traces of information diffusion: they are more likely to be spread from similar sources, by similar people and in similar sequences. Though the diffusion information is pervasively available on social networks, little has been studied due to its special characteristic. Diffusion information refers to by whom and when information is spread, and it is very difficult to directly model. Consider the huge number of social media users and all the possible combinations of spreaders, diffusion information will be of high dimensionality and thus may result in sparsity in the feature space.

Third, while traditional classification tasks mainly focus on optimizing performance metrics like accuracy and F-measure, misinformation detection approaches further take into account the earliness of a method. Social and psychological studies have revealed that misinformation might evolve 70% of its content within 6 transmissions between people (Allport and Postman, 1947). Therefore, ignoring earliness of intervention makes the intervening campaign downgrade rapidly due to the evolved content. The earliness, or timeliness of a method describes how fast can a misinformation detection method be ready to classify misinformation. In this context, the challenge of data scarcity immerses as a key issue of solving the earliness problem. Annotating a dataset could be very time-consuming, and it brings in an unavoidable delay for existing systems, resulting in significant challenges to enable the system to detect new incidents of misinformation in a timely manner.

Fourth, it is increasingly risky to depend on social media data for decision making due to the novel challenges brought by misinformation. The vast amount of online data allow for an insight into the public opinion that has been utilized for predicting the stock price (Bollen *et al.*, 2011) and election results (Tumasjan *et al.*, 2010). Most social media platforms are open to register and easily accessible. For example, thousands of bot accounts were found to intentionally spread misinformation during the 2016 U.S. election[7]. Beyond detecting misinformation, it is appealing if a algorithm such as social recommendation or user profiling methods can adaptively mitigate the negative effect of misinformation.

A key challenge for misinformation-related tasks is that available label information is very scarce. As malicious behaviors usually involve with a lot of disguise, an effective machine learning model needs fine-grained and high quality label information to get training. In order to distinguish such camouflage from a misinformation spreader's content, it is ideal to have a label for each post. However, in real-world, collecting labels for accounts is already challenging, not to mention labels for the posts. With limited label information, existing studies have to assume content of an account is homogeneous and mix all posts together as an atom. Therefore, the malicious content is usually overwhelmed by the vast amount of legitimate content. Similarly, the label information for each link to classify network manipulation is also very difficult to obtain.

To complicate the problem, user feedback can be very biased due to the filter bubble effect (Pariser, 2011). For many other problems, user feedback signals, such as user clicks and reviews, are regarded as a gold standard. In social media platforms,

[7] https://www.nytimes.com/2017/09/07/us/politics/russia-facebook-twitter-election.html

content is being personalized in a framework consisting of two components: model initialization and model update. A typical personalization model is usually updated based on how a user reacts to the content generated by the original model. Therefore, a model actively looks for content that users are more likely to click, and users are more used to clicking certain content. The reinforcement would finally cause a user to get exposed to information that conforms to his/her previous beliefs, regardless of content being true or fake.

In the dissertation, I study the problem of misinformation detection in the context of social media. In order to tackle the challenges, I investigate the following questions,

- How to detect misinformation spreaders and their misbehavior in the presence of adversarial attacks through content and network?
- How to identify useful contextual information, and utilize the descriptive patterns to facilitate the detection of misinformation?
- How to deliver an effective misinformation model at an early stage of misinformation diffusion with the challenge of data scarcity?
- How to mitigate the negative effect of misinformation on a machine learning model when utilizing the contaminated social media data for research?

To answer the research questions, we summarize the main contributions of the dissertation as follows. I summarize the key characteristics of the task of misinformation detection that distance it from classic statistical machine learning problems. In order to tackle the challenges brought by adversarial attacks and lack of labeled data, we abstract patterns from contextual information to facilitate the task of misinformation detection in terms of effectiveness and earliness.

Rest of the dissertation is organized as follows. In Chapter 2, I review the related work. In Chapter 3, I present the proposed adaptive misinformation spreader detection model in the presence of adversarial attacks. In Chapter 4, I discuss a framework for modeling contextual information to facilitate misinformation detection. In Chapter 5, I propose an early detection model to alleviate the data scarcity problem at an early stage of misinformation diffusion. In Chapter 6 and Chapter 7, I propose methods to prevent misinformation from contaminating social media data actively and passively. Chapter 8 concludes the dissertation and suggest future work

Chapter 2

RELATED WORK

Due to the recent incidents of misinformation in social media, many studies have been focusing on the problem of misinformation detection. In this dissertation, I propose novel statistical machine learning methods to tackle emerging challenges. In this section, I introduce several streams of related work.

(1) Misinformation Detection in Traditional Media

The study of misinformation detection can be traced back to 1940s (Allport and Postman, 1945), when psychological studies try to discover the driving forces of misinformation propagation (Allport and Postman, 1947). Various psychological and social science theories have been investigated to understand misinformation circulating among people. For example, a previous study indicates the interplay between reader sentiment and the diffusion of misinformation, implying that an important story is more likely to influence its audience (Anthony, 1973). Similarly, anxiety has also been found to relate to the diffusion of misinformation (Rosnow, 1991), meaning that people are more likely to propagate misinformation that involve with themselves.

(2) Misinformation Detection in Social Media

In order to tackle the emerging challenges brought by social media, researchers apply existing psychological and social science theories. For example, anxiety of social media users have been studied to predict diffusion of misinformation (Oh *et al.*, 2013). The task of misinformation in social media is usually modeled as a classification task (Wu *et al.*, 2016b).Qazvinian *et al.* employ a feature engineering approach to distinguish misinformation from Twitter's content stream (Qazvinian *et al.*, 2011). Observing misinformation posts usually arise inquiries, Zhao *et al.* compile regular expressions to detect topic with concentrated questions (Zhao *et al.*, 2015). Our recent work studies linking distributed discussion snippets to alleviate the cold-start problem (Sampson *et al.*, 2016a). Meanwhile, systems have also been developed to visualize and track known misinformation (McKelvey and Menczer, 2013; Shao *et al.*, 2016). Through representing data via intuitive visualization, experts can observe and understand how misinformation spreads from node to node, so that they are enabled to supervise the learning procedure of misinformation classifiers with their domain knowledge and expertise (Cao *et al.*, 2016; Zhao *et al.*, 2014).

(3) Misinformation Spreader Detection

Existing content-based approaches can generally be categorized into unsupervised and supervised methods. Unsupervised models aim to find content polluters by finding the evidence of abnormality. For example, Lee et al. propose to employ social honeypots to discover polluters (Lee *et al.*, 2010a), which are based on ideas from intrusion detection. Supervised methods assume that content polluters share similar malicious content, which distinguishes them from normal users (Jindal and Liu, 2007). Along the stream of supervised methods, content and other kinds of information has been extensively studied, such as the network structures (Hu *et al.*, 2013; Wang *et al.*, 2012, 2011), sentiment polarities (Ratkiewicz *et al.*, 2011), the frequency of using hashtags and URLs (Benevenuto *et al.*, 2010), morphological features of messages (Thonnard and Dacier, 2011), and behavioral characteristics (Fei *et al.*, 2015; Mukherjee and et al., 2015).

In addition, existing studies also focus on utilizing network information to identify misinformation spreaders. The network modeling methods can generally be divided into three categories, link-based, neighbor-based and group-based. Link-based methods assume links are generally regarded as social trust from other users, and a small number of links might indicate a spammer being fake (Mccord and Chuah, 2011). The underlying assumption is that social media users are carefully connected, which might not be true in the real world. Since users would simply follow back after being followed, social media users with more followees are found to own more followers generally. A revised solution is to compile features such as the ratio of follower/followee (Lee *et al.*, 2010b). However, spammers could follow users incrementally and unfollow those who did not follow back seeminglessly, which is transient and difficult to notice.

(4) Misinformation Diffusion

Information diffusion models are designed to abstract the pattern of information propagation in a network, such as SIR Model (Kermack and McKendrick, 1927), Tipping Model (Centola, 2010), Independent Cascade Model and Linear Threshold Model (Kempe *et al.*, 2003). The diffusion of misinformation is more related to the trust and belief in social networks. The epidemic models, including SIR and IC, assume the infection occurs between an infectious user and a susceptible user with a predefined probability. The probability may increase with more interactions or other contextual conditions. Although the links of a user are independent, a user who has more infectious friends is more likely to be infected. The tipping and LT models also contain a parameter to estimate probability of a user being infected based on the number of activated friends. Generally, given infinite time and an optimal seed set of senders, they assume all users will be infected. However, the diffusion outcome of misinformation is often the global recovery or immunity. Such phenomena reveal that no matter how widespread a piece of misinformation is diffused, some nodes will not be affected and will keep intervening such diffusion (Acemoglu *et al.*, 2010). In addition, traditional information diffusion models can be adapted for the task of misinformation diffusion by adding two user roles, *i.e.,* misinformation diffusers and targeted receivers (Karlova and Fisher, 2013). When receivers receive some information from diffusers, they judge whether to trust and further pass the information based on contextual features. A unified model that jointly considers information and misinformation diffusion is also available (Agrawal *et al.*, 2011).

(5) Social Recommendation

We study robust social recommendation in the presence of misinformation. Recommender systems aim to predict preferences based on prior behaviors, such as purchasing or viewing (Herlocker *et al.*, 2000), or based on the similarity between products and user preferences (Pazzani and Billsus, 2007). Detailed reviews about recommender systems can be accessed in the survey (Bobadilla *et al.*, 2013). Existing social recommendation introduces another two data sources, *i.e.,* social network structures and the user generated content from social media. For example, network-based algorithms, or relational learning, infer the preferences of a user based on its neighbors in the network. Network structures can also be transformed into numerical features (Eldardiry and Neville, 2012; Jensen *et al.*, 2004), and these features are used to learn user preferences. Other approaches try to interpret a user's membership of different social groups (Xu *et al.*, 2008; Neville and Jensen, 2005).

Chapter 3

MISINFORMATION DETECTION OF SPREADERS

In this chapter, I focus on the problem of detecting misinformation spreaders that actively manipulate network structures and content information. The distinct characteristics of social media websites bring about novel challenges for the task. I will introduce the background, formally define the computational problem, and present the proposed method. Based on datasets obtained from real-world social media platforms, we conduct experiments to compare with state-of-the-art approaches.

3.1 Exploratory Study

In this section, we explore the behavioral characteristics of misinformation spreaders. In order for the exploratory study, we manually label posts of known spreaders with a Twitter dataset. We sample 1,500 misinformation spreaders and 1,500 normal users from the large dataset. The annotation is conducted by human annotators on

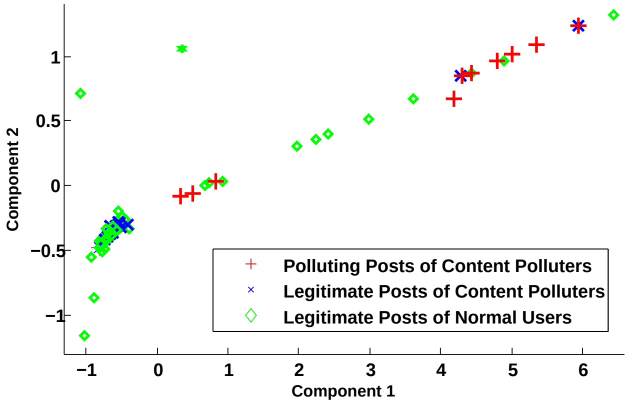


Figure 3.1: Distribution of Posts from Known Misinformation Spreaders and Normal Users with the First Two Principal Components. Many Posts from Misinformation Spreaders Are Legitimate.

Amazon Mechanical Turk[1]. Each post is checked by at least five annotators and the majority label is used. Criteria for the annotators is whether a post violates the Twitter community rules[2]. Therefore we obtain a small collection of post labels for an exploratory study on content.

Content information is usually of high dimensionality that is hard to visualize, so we use the first two principal components of the content with Principal Component Analysis (PCA) to show its distribution. We illustrate three kinds of posts, *i.e.,* legitimate posts of normal users, polluting and legitimate posts of misinformation spreaders, in Figure 3.1. Through observing the figure, we find that many posts of a misinformation spreader are similar to the content of normal users, which manifests camouflage of misinformation spreaders. Traditional approaches merge posts of an account altogether as an attribute vector, which would be less distinguishable to detect camouflaged misinformation spreaders.

3.2 Camouflaged Misinformation Spreaders in Social Media

Internet media continues to pervade our culture, such as social networks, web forums and the blogosphere. The expansive channel for communication facilitates information dissemination between a large group of people. However, motivated by the monetary rewards, misinformation spreaders, which include fraudsters, scammers, and spammers, unfairly overpower normal users by spreading disinformation, which undermines the role of Internet media in sustaining a society as a collective entity.

An emerging characteristic that further complicates the problem is the *camouflage.* Due to the openness of Internet media, it is easy for misinformation spreaders to copy a significant portion of content from normal users. The polluting content that

[1] https://www.mturk.com/

[2] https://support.twitter.com/articles/18311-the-twitter-rules

is camouflaged by the legitimate messages can be very deluding due to the *cognitive inertia*: once many genuine posts from a fraudster establish trust, the fraudulent post is likely to convince many of the readers.

Recent studies have investigated the camouflage of fraudsters from the perspective of network structures (Hooi *et al.*, 2016), proving that network camouflage could be efficiently detected through studying the abnormality of the density of a graph caused by the camouflage links. In this section, we focus on precisely the other side of the problem, *i.e.,* detecting misinformation spreaders in the presence of camouflage. Our goal is to detect polluters under camouflage.

It is particularly difficult and challenging to detect camouflaged misinformation spreaders. Due to the massive amount of content information on Internet media, there is a lack of availability of label information for camouflaged posts. Therefore, traditional fraud and opinion spam detection approaches (Fei *et al.*, 2015; Jiang *et al.*, 2016; Liu, 2012) are not applicable for this problem. In addition, existing work on misinformation spreader detection (Han and Park, 2013; Hu *et al.*, 2014, 2013; Mukherjee and et al., 2015; Wang *et al.*, 2011) only exploits label information for accounts, so they cannot account for the camouflage. Another challenge is data scarcity. Since camouflage can take up the majority of content from a misinformation spreader, it is not easy to identify the scarce polluting evidence, and manually labeling it could be time-consuming and labor-intensive.

In order to tackle the challenges, we propose to utilize label information of accounts. Account labels are easier to obtain and publicly available at a relatively large scale on various platforms, such as social networking sites (Thomas *et al.*, 2011; Webb *et al.*, 2008), the blogosphere (Kolari *et al.*, 2006), and web forums (Niu *et al.*, 2007). Our key intuition is to employ discriminant analysis (Fukunaga, 2013) to capture signals of content pollution with label information of accounts. Motivated by results of recent studies that camouflage tends to be *random* while malicious content is *alike* due to the similar fraudulent targets (Hooi *et al.*, 2016), we assume that the *intersection* of misinformation spreaders' posts in the feature space is more likely to be a signal of polluting content, which can distinguish misinformation spreaders from normal users.

Discriminant analysis has only been studied on the level of features, requiring label information of posts to be available. We make the first attempt to investigate how Camouflaged Content Polluters can be detected with Discriminant Analysis. In particular, we introduce a novel method CCPDA, which effectively detects misinformation spreaders by mining signals of camouflaged pollution.

3.3 Problem Statement

In this section, we introduce the notations used in this section and then formally define the problem we study.

Throughout this section, matrices are denoted as uppercase bold letters (*e.g.*, $\mathbf{V}$), column vectors are denoted as lowercase bold letters (*e.g.*, $\mathbf{c}$) and scalars as lowercase letters (*e.g.*, c). $\mathbf{V}_{i,j}$ denotes the entry at the i^{th} row and j^{th} column of $\mathbf{V}$. $\mathbf{V}_{i,*}$ and $\mathbf{V}_{*,j}$ denotes the i^{th} row and j^{th} column of matrix $\mathbf{V}$, respectively. c_i means the i^{th} element of the column vector $\mathbf{c}$. For any vector $\mathbf{c} \in \mathbb{R}^p$, ℓ_q*-norm* of $\mathbf{c}$ is $||\mathbf{c}||_q = (\sum_{i=1}^{p} |c_i|^q)^{\frac{1}{q}}$ for $q \in (0, +\infty)$. $\langle \mathbf{A}, \mathbf{B} \rangle$ represents $Trace(\mathbf{A}^T\mathbf{B})$. $\mathbf{I}$ is the identity matrix and $\mathbf{1}$ is a vector with all elements to be 1.

Let $\mathbf{A} = [\mathbf{V}, \mathbf{P}, \mathbf{t}]$ be a target account set with post information $\mathbf{V}$, user-post mapping $\mathbf{P}$ and identity labels $\mathbf{t}$. The data matrix $\mathbf{V} \in \mathbb{R}^{m \times n}$ is the post information of all users, where m is the number of posts and n is the number of textual features extracted from the posts. We denote the user-post association as $\mathbf{P} \in \mathbb{R}^{u \times m}$, where u is the number of users. $\mathbf{P}_{i,j}$ equals to 1 if the j^{th} post is posted by the i^{th} account and equals to 0 otherwise. $\mathbf{t} \in \{0,1\}^{u \times 1}$ records the identity label of all users, where $\mathbf{t}_i = 1$ represents the i^{th} account is a misinformation spreader.

We now define the problem of detecting camouflaged misinformation spreaders as follows:

Given a set of accounts $\mathbf{A}$ *with post information* $\mathbf{V}$*, user-post mapping matrix* $\mathbf{P}$*, and identity label information* $\mathbf{t}$ *for partial accounts, our goal is to learn a model with the best performance to classify whether a user is a misinformation spreader.*

3.4 Detecting Camouflaged Misinformation Spreaders

In this section, we introduce our model CCPDA and present the efficient optimization algorithm. In the end, we theoretically discuss the time complexity and its scalability in real applications.

3.4.1 Modeling Content Information

We represent posts with a data matrix $\mathbf{V} \in \mathbb{R}^{m \times n}$, where each row represents a post and each column represents a textual feature. However, since labels of posts are unavailable, we start with a trivial solution that labels all posts of a known polluter as polluting. Then it can be reduced to the least square problem (Lawson and Hanson, 1974):

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{V}\mathbf{w} - \mathbf{y}||_2^2 + \frac{\lambda_1}{2}||\mathbf{w}||_2^2, \tag{3.1}$$

where $\mathbf{w} \in \mathbb{R}^n$ is the model to learn and a regularization term $||\mathbf{w}||_2^2$ is imposed to avoid overfitting. The parameter λ_1 controls the extent of the model complexity. The vector $\mathbf{y} \in \mathbb{R}^m$ is the pseudo label that is temporally initialized. The pseudo label vector can be derived from the account labels $\mathbf{y} = \mathbf{P}^T\mathbf{t}$. However, as investigated in Section 3.1, posts of misinformation spreaders are not necessarily fraudulent, so labeling all posts of a misinformation spreader as positive would make the classifier lose sensitivity to content pollution and result in a low *recall*. Next, we discuss how we incorporate discriminant analysis to solve this problem.

3.4.2 Detecting Camouflaged Polluters with Discriminant Analysis

Preliminarily, we label posts in a trivial manner. In order to enable the modification of the label values, we introduce a weighting vector $\mathbf{c} \in \mathbb{R}^m$ for the label vector. Through incorporating the weight, the label of the i^{th} post becomes $c_i y_i$. So the labels could be updated through updating the weights. Our aim is to filter out camouflage, *i.e.*, increasing weights of polluting posts and decreasing weights of labels of polluters' legitimate content. To this end, we reformulate the objective function in Eq.(3.1) as:

$$\min_{\mathbf{w},\mathbf{c}} \frac{1}{2}\sum_{i=1}^{m}(c_i y_i - \mathbf{V}_{i,*}\mathbf{w})^2 + \frac{\lambda_1}{2}||\mathbf{w}||_2^2, \tag{3.2}$$

where c_i represents the weight of i^{th} post. Since the normal posts of a misinformation spreader are initially labeled as positive, which can be viewed as mislabeled examples, they are more likely to cause a larger reconstruction error during training (Hawkins and et al., 2002). Therefore, penalizing large errors leads to downweighting labels of legitimate content. In addition, since labels of legitimate users are of value 0, the weight does not influence normal users during the optimization.

Since content pollution may only comprise a small portion of all posts, the representation of $\mathbf{c}$ should be sparse. Motivated by sparse representation learning (Ng, 2004), where only few *coefficients* are assumed to reveal the key information, we incorporate an ℓ_1-*norm* with $\mathbf{c}$ and reformulate the objective function as follows:

$$\min_{\mathbf{w},\mathbf{c}} \frac{1}{2}\sum_{i=1}^{m}(c_i y_i - \mathbf{V}_{i,*}\mathbf{w})^2 + \frac{\lambda_1}{2}||\mathbf{w}||_2^2 + \lambda_2||\mathbf{c}||_1, \tag{3.3}$$

where the ℓ_1-*norm* penalizes non-sparse solutions. The parameter λ_2 controls the extent of sparsity, which can be regarded as the discriminant threshold for a post to be selected and labeled as polluting. Through introducing the sparsity regularizer, the selected entries are likely to be 1 while the unselected entries are likely to be exactly zero (Ng, 2004), which is favorable since a post is either fraudulent or legitimate in real-world applications.

Sparse representation methods are used to find dominant signals (Donoho and Elad, 2003). In traditional studies, the dominance is determined by *frequency*, meaning that, in the context of misinformation spreader detection, the polluting content that appears most frequently would be more likely to be selected. This is helpful for finding content pollution since malicious content is usually similar. However, it also results in some polluters to be overlooked. If some misinformation spreaders are involved with a *smaller* campaign, and the “frequency” does not “exceed” the discriminant threshold, posts of all these polluters would be disregarded. Therefore, the polluting information that is useful for identifying future polluters would also be ignored. In order to fully exploit the label information, we force every polluter to be selected with some posts by introducing an $\ell_{1,2}^{\mathcal{G}}$-*norm* term. The regularization term is as follows,

$$\ell_{1,2}^{\mathcal{G}}(\mathbf{c}) = \sum_{g \in \mathcal{G}} ||\mathbf{c}_{\mathcal{G}_g}||_1^2. \tag{3.4}$$

The $\ell_{1,2}^{\mathcal{G}}$-*norm*, which is also called group exclusive penalty (Kong *et al.*, 2014), is proposed to select discriminant features of different groups. Here, $\mathcal{G}$ is the set of all groups, where $\mathcal{G}_g$ denotes the indices of posts in a group $g \in \{1, 2, \ldots, m\}$. For example, let $\mathcal{G}_g = \{1, 2, 4, \cdots\}$, then $||\mathbf{c}_{\mathcal{G}_g}|| = [c_1, c_2, 0, c_4, 0, \ldots, 0]$. The $\ell_{1,2}^{\mathcal{G}}$-*norm* first sums up absolute values of intra-group variables and then imposes an ℓ_2-*norm* to regularize the sum. The minimization process leads to intra-group sparsity. Concretely, it enforces locally discriminant posts of a polluter to be upweighted while enforces globally discriminant content to be downweighted.

The group exclusive penalty is convex but non-smooth, which is difficult for optimization. In order to solve the problem, we rewrite the $\ell_{1,2}^{\mathcal{G}}$-*norm* as follows,

$$\begin{aligned}
\ell_{1,2}^{\mathcal{G}_g}(\mathbf{c}) &= \frac{1}{2}\sum_{i=1}^{u}(\mathbf{c}^T\mathbf{P}_{i,*})^2 && (3.5)\\
&= \frac{1}{2}\sum_{i=1}^{u}\mathbf{c}^T\mathbf{P}_{i,*}^T\mathbf{P}_{i,*}\mathbf{c} && (3.6)\\
&= \frac{1}{2}\mathbf{c}^T\mathbf{P}^T\mathbf{P}\mathbf{c}, && (3.7)
\end{aligned}$$

where $\mathbf{P}$ denotes the user-post mapping matrix. Since $\mathbf{P}$ is a constant matrix, we introduce $\mathbf{M} = \mathbf{P}^T\mathbf{P}$ to replace the product. In $\mathbf{M} \in \mathbb{R}^{m\times m}$, $M_{i,j}$ equals to one if post i and j are generated by the same user and zero otherwise. By rewriting the regularization term $\ell_{1,2}^{\mathcal{G}_g}(\mathbf{c})$, it is convex and smooth, which can be easily incorporated into the objective function in Eq.(3.3) as:

$$\frac{1}{2}\sum_{i=1}^{m}(c_iy_i - \mathbf{V}_{i,*}\mathbf{w})^2 + \frac{\lambda_1}{2}||\mathbf{w}||_2^2 + \lambda_2||\mathbf{c}||_1 + \frac{\lambda_3}{2}\mathbf{c}^T\mathbf{M}\mathbf{c}, \qquad (3.8)$$

where the parameter λ_3 controls the importance of locally discriminant content.

Since we model individual posts, the resultant data size should be very large. In order to tackle the challenge of *big data*, we introduce an optimization algorithm that can optimize the problem in Eq.(4.5) efficiently.

3.4.3 An Optimization Algorithm

The optimization problem in Eq.(4.5) is not jointly convex with respect to the two variables $\mathbf{w}$ and $\mathbf{c}$ together. However, by fixing one of them, the objective function is convex to the other. So we propose to find optimal solutions through alternatively updating one by fixing the other.

1) *While fixing* $\mathbf{c}$, *update* $\mathbf{w}$: the problem only depends on $\mathbf{w}$. By only considering

items related to $\mathbf{w}$, we reformulate the objective as follows:

$$\epsilon_{\mathbf{w}} = \frac{1}{2}\sum_{i=1}^{m}(c_i y_i - \mathbf{V}_{i,*}\mathbf{w})^2 + \frac{\lambda_1}{2}||\mathbf{w}||_2^2, \tag{3.9}$$

which is reduced to an ℓ_2 regularized linear regression problem. In order to cope with the massive amount of content information, we adopt Stochastic Gradient Descent (SGD) (Bottou, 2010) to solve the optimization problem. SGD belongs to a class of hill-climbing optimization technique that seeks a stationary point of a function. To utilize SGD, we derive the gradient of $\mathbf{w}$ as follows:

$$\frac{\partial \epsilon_{\mathbf{w}}}{\partial \mathbf{w}} = \sum_{i=1}^{m}\mathbf{V}_{i,*}^T(\mathbf{V}_{i,*}\mathbf{w} - c_i y_i) + \lambda_1 \mathbf{w}. \tag{3.10}$$

Instead of updating in a batch mode, SGD randomly selects data examples from the total m data instances. The update process can then be significantly accelerated with the multi-threading manner. A detailed discussion about the performance of single- and multi-thread implementations of the optimization algorithm is presented later in Section 3.4.4.

Therefore, the optimal predictor can be achieved through the following update rules:

$$\mathbf{w} = \mathbf{w} - \tau\frac{\partial \epsilon_{\mathbf{w}}}{\partial \mathbf{w}}, \tag{3.11}$$

where τ is a learning rate which we set using backtracking line search (Armijo, 1966).

2) *While fixing* $\mathbf{w}$, *update* $\mathbf{c}$: the problem only depends on $\mathbf{c}$. Since the reconstruction $\mathbf{V}_{i,*}\mathbf{w}$ becomes constant, we use $\mathbf{e}$ to replace it, where $e_i = \mathbf{V}_{i,*}\mathbf{w}$. Thus, Eq.(4.5) can be reformulated as follows:

$$\min_{\mathbf{c}} \frac{1}{2}\sum_{i=1}^{m}(c_i y_i - e_i)^2 + \lambda_2||\mathbf{c}||_1 + \frac{\lambda_3}{2}\mathbf{c}^T\mathbf{M}\mathbf{c}. \tag{3.12}$$

Though all components in Eq.(3.12) are convex with respect to $\mathbf{c}$, the ℓ_1-*norm* makes it non-smooth, which is difficult to optimize. Following (Liu *et al.*, 2009), we try to optimize the problem in Eq.(3.12) through reformulating it as an equivalent smooth and convex problem.

Theorem 1 *Eq.(3.12) is equivalent to the following ℓ_1-ball constrained smooth convex optimization problem:*

$$\begin{aligned} &\underset{\boldsymbol{c}\in\mathcal{Z}}{min} \quad \mathcal{O}(\boldsymbol{c}) = \frac{1}{2}||\boldsymbol{c}\circ\boldsymbol{y}-\boldsymbol{e}||_2^2 + \frac{\lambda_3}{2}\boldsymbol{c}^T\boldsymbol{M}\boldsymbol{c}, \\ &where \quad \mathcal{Z} = \{\boldsymbol{c} \mid ||\boldsymbol{c}||_1 \le z\}. \end{aligned} \tag{3.13}$$

and $\circ$ denotes component-wise multiplication. $z \geq 0$ is the radius of the ℓ_1-ball. λ_2 and z have a 1:1 correspondence.

Since $||\mathbf{c}||_1$ is a valid norm, it is a closed convex function. It defines a closed and convex set $\mathcal{Z}$ (Note that $\mathcal{Z}$ is not empty, since $z > 0$ and zero matrix belongs to $\mathcal{Z}$). The second derivative of $\mathcal{O}$ (The Hessian matrix) is symmetric and positive semi-definite, so (3.13) is convex and differentiable. $\mathcal{O}$ is a convex and differentiable function in a closed and convex set $\mathcal{Z}$, which is equivalent to the problem in Eq.(3.12).

The ℓ_1-*ball* constrained convex problem in Eq.(3.13) can be efficiently solved. Motivated by (Ji and Ye, 2009), we adopt proximal gradient descent. The update rule for $\mathbf{c}$ can be formulated as follows:

$$\mathbf{c}^t = \arg\min_{\mathbf{c}\in\mathcal{Z}} P_{\gamma,\mathbf{c}^{t-1}}(\mathbf{c}), \tag{3.14}$$

where the superscript t denotes the number of iteration, and $P_{\gamma,\mathbf{c}^{t-1}}(\mathbf{c})$ is the convex problem's Euclidean projection onto the constraint space (Boyd and Vandenberghe, 2004). The projection can be formulated as follows,

$$P_{\gamma,\mathbf{c}^{t-1}}(\mathbf{c}) = \mathcal{O}(\mathbf{c}^{t-1}) + \langle\nabla\mathcal{O}(\mathbf{c}^{t-1}), \mathbf{c}-\mathbf{c}^{t-1}\rangle + \frac{\gamma}{2}||\mathbf{c}-\mathbf{c}^{t-1}||_2^2, \tag{3.15}$$

where $\nabla\mathcal{O}(\cdot)$ is the derivative of $\mathcal{O}(\cdot)$. Since $\mathcal{O}(\cdot)$ is convex, $\nabla\mathcal{O}(\cdot)$ can be derived from Eq.(3.13) as

$$\nabla\mathcal{O}(\mathbf{c}) = \mathbf{y}\circ\mathbf{c}\circ\mathbf{y} - \mathbf{y}\circ\mathbf{e} + \lambda_3\mathbf{M}\mathbf{c}. \tag{3.16}$$

Given a problem in the form of Eq.(3.15), the analytical solution can be directly obtained (Ji and Ye, 2009). The solution of $\mathbf{c}$ can be written as

$$c_j^t = max(0, u_j^{t-1}(1 - \frac{\lambda_3}{\gamma |u_j^{t-1}|})), \tag{3.17}$$

where $\mathbf{u}^t = \mathbf{c}^t - \frac{1}{\gamma}(\nabla \mathcal{O}(\mathbf{c}^t))$, which is introduced to replace the gradient step, and u_j^t and c_j^t are the j^{th} element of $\mathbf{u}^t$ and $\mathbf{c}^t$, correspondingly.

The detailed algorithm of CCPDA is presented in Algorithm 6. $\mathbf{w}$ is updated in line 3 and $\mathbf{c}$ is updated in line 13. From line 4 to line 12, Goldstein-Armijo line search schemes (Armijo, 1966) are adopted to find an optimal γ.

3.4.4 Time Complexity and Scalability

Here we analyze the time complexity of the algorithm to solve the objective function in Eq.(4.5). The computational cost for $\mathbf{w}$ depends on the constrained linear regression in Eq.(3.9), which is $O(m^2n+mn^2)$. The computational cost for $\mathbf{c}$ depends on the calculation of the Euclidean projection, which can be analytically solved in $O(m)$. In real applications the number of posts m is usually large, while the number of features n can usually be reduced by feature selection, so the computational cost of solving Eq.(4.5) is dominantly determined by m. Posts are usually short and sparse, meaning that many of them are independent. Therefore, SGD can be employed in a parallel manner to increase the speed. We introduce more details about the model scalability in Section 3.5.5.

3.5 Experiments

In this section, we conduct experiments to evaluate the effectiveness and efficiency of the proposed framework. Through the empirical studies, we aim to answer the following three questions:

Algorithm 1 Optimization Algorithm for CCPDA

```
Input: $\{\mathbf{V}, \mathbf{y}, \mathbf{P}, \mathbf{w}^0, \mathbf{c}^0, \lambda_1, \lambda_2, \lambda_3, \gamma^0, max_{iter}\}$
Output: $\mathbf{w}$
Initialize $\mathbf{w}^1 = \mathbf{w}^0$, $\mathbf{c} = \mathbf{c}^0$, $t = 1$
while Not convergent and $t \leq max_{iter}$
    Update $\mathbf{w}$ with Eq.(7.8)
    Set $\gamma = \gamma^0$
    loop
        Calculate $\mathbf{c}^t$ with $\mathbf{c}^{t-1}$, $\gamma$ and Eq.(3.17)
        If $\mathcal{O}(\mathbf{c}^t) \leq \mathcal{P}_{\gamma,\mathbf{c}^{t-1}}(\mathbf{c}^t)$ then
            $\gamma = \gamma/2$
            break
        end if
        $\gamma = 2 \times \gamma$
    end loop
    Update $\mathbf{c}^t$ with $\mathbf{c}^{t-1}$, $\gamma$ and Eq.(3.17)
    $t = t + 1$
end while
```

- How effective is the proposed approach compared with other methods of misinformation spreader detection?

- What are the effects of discriminant analysis on detecting camouflaged misinformation spreaders?

- How efficient can the proposed approach process large number of users and posts?

We begin by introducing the two real-world datasets and compare CCPDA with

Table 3.1: Statistics of the Dataset Used in This Study.

Posts	Reposts	Unique Users	Positive Ratio
1,150,192	576,167	94,535	7.5%

several competitive methods for detecting misinformation spreaders. Then we study effects of discriminant analysis with regard to precision and recall. Finally, we present the performance of CCPDA with the single-thread and multi-thread implementations.

3.5.1 Dataset

We employ two real-world Twitter datasets. Since over 200 million posts are posted per day on Twitter[3], the popularity has made Twitter a testbed for content pollution research (Hu *et al.*, 2014; Yang *et al.*, 2012). We aim to collect a large dataset that includes posts about all prior polluting content within a certain period. Therefore, we collect the first dataset (TwitterS). A small sample of TwitterS has been used for the exploratory study in Section 3.1.

Existing studies obtain normal accounts through random sampling (Hu *et al.*, 2013; Thomas *et al.*, 2011), where the cutoff between positive and negative examples may not be reflective of the original data. In order to keep in line with the real world distribution, we build up a dataset by randomly crawling all accounts under certain topics, where labels are obtained using the gold standard (Thomas *et al.*, 2011). In particular, we randomly sample posts from Twitter in 2013. In May 2016, we crawl each user in the dataset again and check the account status. We examine the account status via the statuses/user-timeline API endpoint. The status can take on one of three values:

- **Active:** The account is still open on the site, which is regarded as a normal

[3] https://blog.twitter.com/2011/200-million-tweets-per-day

account.

- **Suspended:** The account has been suspended for violating Twitter's policies. This is considered a temporary ban, where the user can petition Twitter to have the account reinstated.

- **Deleted:** The account has been deleted for violating Twitter's community rules. This is considered a permanent ban.

The labels are obtained by using Twitter's APIs to retrieve the response code for each account (Kumar *et al.*, 2014). Among the accounts, we discover that 92.5% of the accounts are active, 4.7% are deleted and the rest are suspended. We consider active accounts as normal users and the rest as misinformation spreaders in this section according to conventional settings (Thomas *et al.*, 2011). Statistics on the dataset are shown in Table 3.1.

The second dataset (TwitterH) is labeled from followers of honeypot accounts (Lee *et al.*, 2014). Honeypots are social media accounts that are created for collecting evidence of misinformation spreaders. In particular, the honeypots only post completely randomized content to attract misinformation spreaders who do not care about the content quality. The accounts that interact (such as retweeting, commenting and following) with honeypots are thus detected as misinformation spreaders.

TwitterS will be made publicly available upon acceptance, and TwitterH is already publicly available[4]. The TwitterH dataset used in our study consists of 11,470 users in total, with a 50% cutoff between the positive and negative data.

[4] http://infolab.tamu.edu/data/

Table 3.2: Results on TwitterH Dataset.

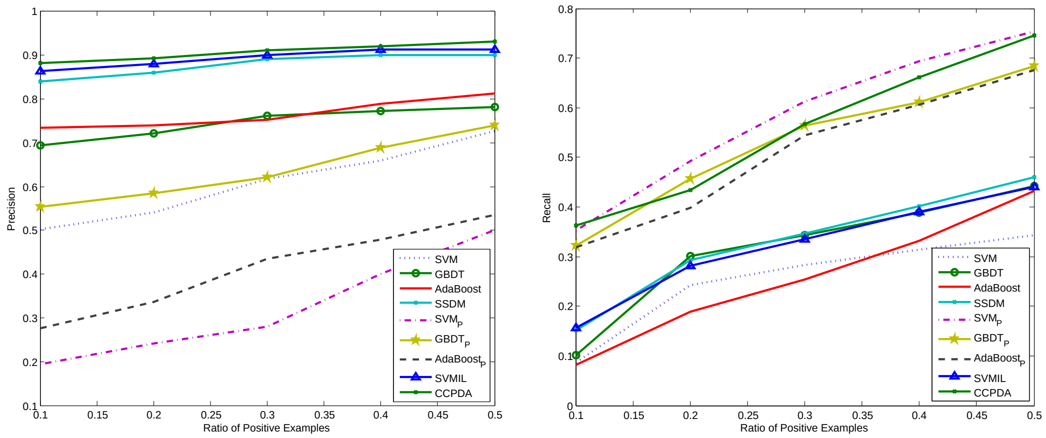
Method	Precision	Recall	F-score
SVM	79.64%	72.47%	75.88%**
GBDT	88.24%	84.26%	86.20%**
AdaBoost	81.35%	69.26%	74.82%**
SSDM	90.87%	83.94%	87.27%*
SVM$_P$	76.33%	88.53%	81.97%**
GBDT$_P$	84.07%	88.47%	86.21%**
AdaBoost$_P$	76.54%	87.57%	81.68%**
SVMIL	89.46%	81.65%	85.38%*
CCPDA	91.20%	88.55%	**89.86%**

Symbol * indicates that CCPDA outperforms a given baseline by 0.05 statistical significance level, ** indicates 0.01.

Table 3.3: Results on TwitterS Dataset.

Method	Precision	Recall	F-score
SVM	29.24%	8.78%	13.53%**
GBDT	69.51%	5.12%	9.66%**
AdaBoost	73.41%	8.24%	14.82%**
SSDM	88.01%	10.11%	18.14%**
SVM$_P$	18.53%	62.14%	28.54%**
GBDT$_P$	55.34%	32.22%	40.72%**
AdaBoost$_P$	27.62%	31.85%	29.59%**
SVMIL	84.36%	13.56%	23.36%**
CCPDA	89.17%	30.26%	**45.96%**

The symbol * indicates that CCPDA outperforms a given baseline by 0.05 statistical significance level, ** indicates 0.01.



(a) Precision of different misinformation spreader detection approaches with an increasing ratio of misinformation spreaders.

(b) Recall of different misinformation spreader detection approaches with an increasing ratio of misinformation spreaders.

Figure 3.2: Precision and Recall of Different Methods with Varying Ratio of Positive Examples by Randomly Down-sampling Normal Users.

3.5.2 Settings

In this section, we test the performance with respect to precision, recall and F-score. In order to investigate the effectiveness of modeling individual posts with discriminant analysis, we include two kinds of baselines: account-centric and post-centric methods. Account-centric methods conventionally construct an attribute vector from all posts of a user. Post-centric methods model a user by learning individual posts. The 10-fold cross-validation is employed to generate all experimental results. We list all baseline methods below,

- Support Vector Machines (SVM) are supervised learning tools for solving binary classification, which have been successfully applied to various tasks.

- AdaBoost is a general boosting framework. It builds up classifiers by ensembling weak classifiers.

- Gradient Boosted Decision Tree is a boosting algorithm which produces a prediction model in the form of an ensemble of multiple decision trees.

- SVMIL belongs to Multiple Instance Learning (MIL) algorithms which extends SVM in a multi-instance setting. MIL shares a similar formulation with our work, assuming that each example contains multiple instances (Zhou, 2004). We report the best result among all available algorithms in the MIL toolkit (Tax, 2015).

- Social Spammer Detection in Microblogging: Hu et al. proposed a framework SSDM to detect misinformation spreaders in social media by jointly modeling network and content information (Hu *et al.*, 2013).

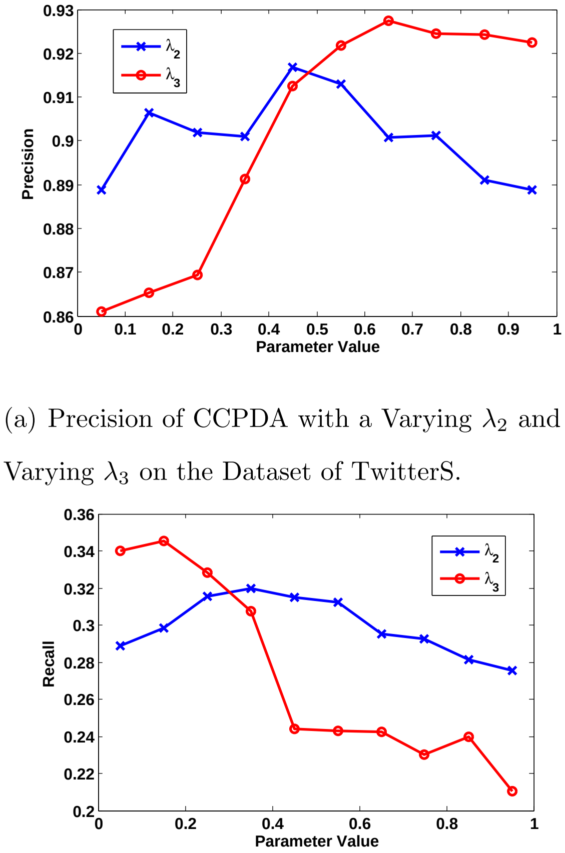
- Post-Centric methods are trained with individual posts. The method is then named by adding a subscript of P to that of corresponding account-centric models, such as SVM$_P$, GBDT$_P$ and AdaBoost$_P$. Since post labels are not available, known misinformation spreaders' posts are all labeled as positive conventionally (Markines *et al.*, 2009).

For post-centric methods including CCPDA, a single detected polluting post leads a user to be classified as positive. For account-centric methods, content of a user is merged into an attribute vector. Parameters of all methods are tuned via cross-validation with a separate validation set.

3.5.3 Experimental Results

The results on two datasets are summarized in Table 3.2 and Table 3.3. Based on the experimental results, we make the following observations:

1) Post-centric methods achieve better recall while account-centric methods achieve better precision. Since an individual suspicious post causes an account to be classi-



(a) Precision of CCPDA with a Varying λ_2 and a Varying λ_3 on the Dataset of TwitterS.

(b) Recall of CCPDA with a Varying λ_2 and a Varying λ_3 on the Dataset of TwitterS.

Figure 3.3: Precision and Recall of CCPDA with Varying Parameters.

fied as a misinformation spreader, post-centric methods are more likely to detect more polluters, which results in the higher recall. By mixing all content together, account-centric approaches focus on the apparent misinformation spreaders, so it results in a higher precision.

2) As shown in Table 3.2, SSDM achieves the second best F-score, showing that jointly exploiting network and information is effective. CCPDA focuses on content information, which can be extended with additional information sources such as the networks.

3) As shown in Table 3.3, GBDT$_P$ achieves the second best F-score by capturing

approximately $\frac{1}{3}$ misinformation spreaders with a 55% precision. Since TwitterS dataset is more skewed, post-centric methods get better F-score by labeling more misinformation spreaders.

4) SVMIL performs better on precision while worse on recall. The basic assumption of multi-instance learning that positive bags share similar instances leads the model to focus more on obvious polluting content, and thus it loses the sensitivity to misinformation spreaders with locally discriminant polluting evidence.

5) CCPDA outperforms all the baselines with respect to F-score. A precision of 91.20% and recall of 88.55% are achieved on the TwitterH dataset. In looking into the results of other post-centric approaches, we find that they are oversensitive and have a lower precision. With discriminant analysis, CCPDA achieves a higher precision by focusing on only the polluting content.

6) We find that all methods perform better on the TwitterH dataset. This is caused by the data skewness. The cutoff between positive and negative examples of the TwitterH dataset is almost 1:1, while only 7% in TwitterS are misinformation spreaders.

Sensitivity to data skewness: In order to test the sensitivity with data skewness, we randomly downsample negative examples to make TwitterS more balanced. We report results of different methods in Figure 3.2. It can be seen that the precision of CCPDA is stable with regard to the change of distribution. SVM$_P$ achieves better recall, while its precision falls behind that of CCPDA.

CCPDA outperforms all baseline methods in terms of F-score on real-world data with different cutoffs. Next, we will further investigate to what extent discriminant analysis facilitates CCPDA.

3.5.4 Importance of Discriminant Analysis

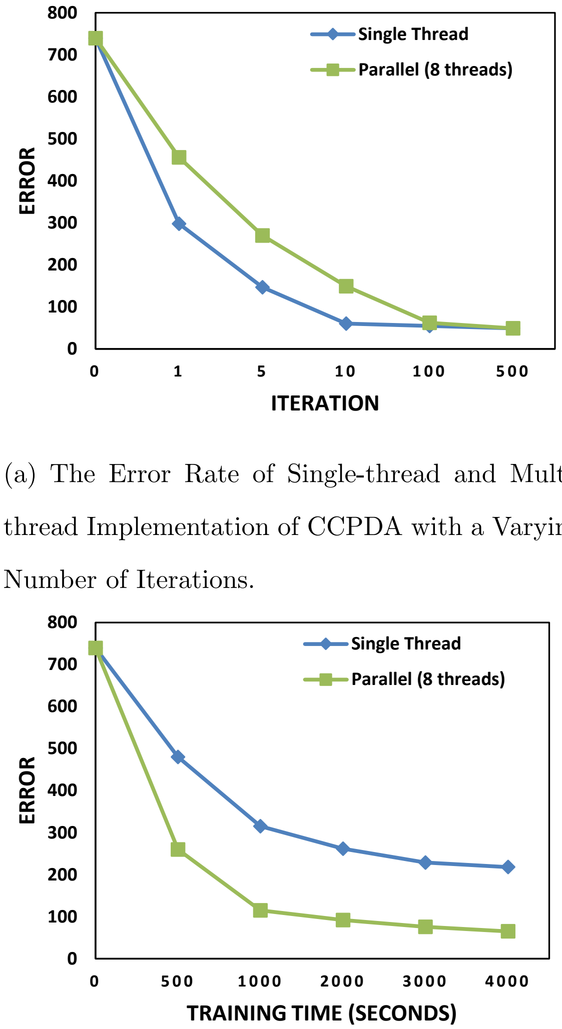
In this subsection, we investigate the impact of discriminant analysis on polluter detection. TwitterS dataset is used for the experiments. In order to visualize the effect caused by discriminant analysis, we show the precision and recall by varying λ_2 and λ_3 in Figure 3.3. Note that λ_2 controls the global discriminant threshold and λ_3 controls local discriminant threshold. First, we notice that, for λ_2 the best precision is achieved when the value is around 0.35 to 0.45, and the best recall is achieved around 0.3 to 0.4, which indicates that focusing on the top polluting few posts ignores useful information, while focusing on too many would lead the discriminant posts to be overwhelmed by the rest. Second, with increasing λ_3, precision and recall increases and decreases almost monotonously, which indicates that it controls the trade-off between accuracy and sensitivity. When λ_3 is large, the most discriminant posts are selected, which results in the model to be more precise. Meanwhile, when λ_3 is small, the local discriminant posts are higher weighted which results in a higher recall.

The results show that CCPDA leverages the sparse structure of posts to find discriminant content. λ_2 and λ_3 control the balance between precision and sensitivity.

3.5.5 Performance Analysis

In practice, Internet media such as Twitter and Facebook usually contain a large number of users and posts. In order to cope with the scalability challenge, we employ parallel SGD with 8 threads to optimize the model. In this section, we evaluate the performance of two methods by measuring the convergence speed (training error) of single-thread SGD and parallel SGD with regard to the number of iterations and the amount of time.

Figure 3.4(a) shows the training error with varying number of iterations. The



(a) The Error Rate of Single-thread and Multi-thread Implementation of CCPDA with a Varying Number of Iterations.

(b) The Error Rate of Single-thread and Multi-thread Implementation of CCPDA with the Varying Time for Training.

Figure 3.4: Comparison of Training Convergence Speed of Single Thread and Parallel Learning in Terms of the Number of Iterations (a) and Time in Seconds (b). Though Multi-thread SGD Converges with More Iterations, Parallel Optimization Significantly Reduces the Training Time.

training error of single-thread SGD decreases faster than that of the parallel method. After approximately 100 iterations, the training error of parallel SGD converges to that of the single-thread SGD with 10 iterations. Since posts are usually short and the content is sparse, parallel updates can significantly accelerate the speed of training. Figure 3.4(b) shows the training error with varying time. Given the same time (4,000 seconds), parallel SGD has run over 30 iterations and achieved a much lower training error, while single-thread SGD has only run for fewer than 5 iterations. The experimental results show that CCPDA converges fast and it can efficiently cope with real-world data at a large scale with multi-threading.

3.6 Summary

Camouflage of misinformation spreaders presents great challenges to Internet media. In this section, we investigate how the camouflaged polluting signal can be identified with label information only for accounts. In particular, the proposed framework utilizes discriminant analysis to discover the key post that distinguishes misinformation spreaders. Also, we present an efficient algorithm to solve the proposed nonsmooth convex optimization problem. Experimental results on real-world Twitter datasets demonstrate that the proposed framework can effectively utilize available information to outperform the state-of-the-art approaches. There are many potential future extensions of this section. First, it would be interesting to jointly consider camouflage in other online activities, such as social network structures and user profiles, for misinformation spreader detection. Also, polluting strategies may evolve over time, so it would be useful to explore incremental update rules with streaming data.

Chapter 4

MISINFORMATION DETECTION WITH CONTEXTUAL INFORMATION

In this chapter, I focus on the problem of misinformation detection with contextual information. I will first review the background and introduce how contextual information can help expose misinformation. In addition, I will introduce the problem formulation and present the proposed framework. Real-world data obtained from a social media platform has been used to evaluate the proposed method against the state-of-the-art approaches.

4.1 Misinformation in Social Media: Content and Context

As online social networks continue to pervade our culture, social networking sites have become an attractive platform to facilitate the spread of information. A recent study from Pew Research claims that 62% of adults get their news from social media in United States, with 29% among them doing so very often[1]. Concomitant with the expansive and varied sources of data are the challenges for personalizing the massive amount of information and filtering out unwanted messages such as fake news and spam. However, the sparse and noisy social media content makes it difficult for traditional approaches, which heavily rely on content features, to tackle these challenges.

By contrast, our study aims to find additional data sources to solve the problem. In this section, we focus on the diffusion of information. A key driving force behind the diffusion of information is its spreaders. People tend to spread information that caters to their interests and/or fits their system of belief (Del Vicario *et al.*, 2016).

[1] http://www.journalism.org/2016/05/26/news-use-across-social-media-platforms-2016/

Hence, similar messages usually leads to similar traces of information diffusion: they are more likely to be spread from similar sources, by similar people and in similar sequences. Since the diffusion information is pervasively available on social networks, in this section, we aim to investigate how the traces of information diffusion in terms of spreaders can be exploited to categorize a message. The message can be a piece of news, a story or a meme that has been posted and forwarded in social networks, and those users who post or forward it are the spreaders. Traces of a message refer to by whom and when the message is spread, *i.e.*, posted or forwarded.

We propose TraceMiner, a novel approach for classifying social media messages with diffusion network information. TraceMiner takes traces of a message as input and outputs its category. Consider the huge number of social media users and all the possible combinations of spreaders, traces will be of high dimensionality and thus may result in sparsity in the feature space. To cope with the problem, TraceMiner utilizes the proximity of nodes (Tang *et al.*, 2015) and social dimensions (Tang and Liu, 2009) manifested in the social network, which have been successfully applied to capture the intrinsic characteristics of social media users in a myriad of applications.

To demonstrate TraceMiner's potential on real-world applications, we evaluate it with traditional approaches on Twitter data. TraceMiner outperforms competitors on multi-label information classification problems in large graphs. Therefore, TraceMiner provides an alternative way for modeling social media messages through learning abundant diffusion data that has not be fully utilized. Existing graph mining research mainly focuses on learning representation of graphs and nodes, while little attention has been paid to classifying information circulating between nodes. TraceMiner distances from existing graph representation methods by directly modeling information and making predictions in an end-to-end manner other than providing only an attribute vector or embedding vector.

4.2 Problem Statement

We consider the problem of classifying social media messages into one or more categories. We define a graph $G \in \langle V, E \rangle$, where $v_i \in V$ with $i \in [1, |V|]$ is a node (user) and $E \subseteq V \times V$ is the set of edges. If $e_{ij} \in E$, there is an edge between v_i and v_j, otherwise there is not. Let M be the set of messages where $m_i \in M$ with $i \in [1, |M|]$. Each message m_i has a corresponding set of spreaders $\{(v_1^{m_i}, t_1^{m_i}), \{(v_2^{m_i}, t_2^{m_i}), \cdots, \{(v_n^{m_i}, t_n^{m_i})\}$, where n is the number of spreaders for m_i and $v_j^{m_i}$ is a user who spreads m_i at the time of $t_j^{m_i}$. Messages are partially labeled and thus only some of them have an associated class label. We denote the set of labels as Y, where $y_i \in Y$ indicates that m_i is labeled. Our goal is to learn a model with the social network graph G and partially labeled message M with the corresponding diffusion traces and label information Y, to predict $\hat{y}$ for the unlabeled messages.

Problem definition for traditional approaches: In order to make predictions for messages, most existing methods take the problem as a text categorization task, hence, each message m_i has a set of spreaders $\{(v_1^{m_i}, t_1^{m_i}, c_1^{m_i}), \cdots, \{(v_n^{m_i}, t_n^{m_i}, c_n^{m_i})\}$, where $c_j^{m_i}$ is the content information.

4.3 Exploiting Context in Detecting Misinformation

In this section, we introduce how a diffusion trace can be used to facilitate classification. We first utilize sequential modeling methods to enable sequences to be used as attribute vectors. To alleviate the sparsity of sequences, we present a novel embedding method.

4.3.1 Sequence Modeling

Given the spreader information $\{(v_1^{m_i}, t_1^{m_i}), \cdots, \{(v_n^{m_i}, t_n^{m_i})\}$ and the graph G, the topology of information diffusion can be inferred by graph mining techniques (Gomez Rodriguez *et al.*, 2010). The topology, which is usually a tree or forest (multiple trees) rooted with the initial spreader, contains informative patterns for characterizing a message. However, it is extremely difficult to directly deal with the tree structure. Consider two messages with similar diffusion networks, adding or removing one spreader, or changing any direction of the information flow would lead to a different tree. Theoretically, there can be n^{n-2} different trees with n number of different nodes according to the Cayley's formula (Clarke, 1958).

In order to solve this problem, we convert the tree structure into a temporal sequence. For example, given the spreaders of m_i $\{(v_1^{m_i}, t_1^{m_i}), \cdots, \{(v_n^{m_i}, t_n^{m_i})\}$, we generate a sequence $x_i = [(v_{q(1)}^{m_i}, t_{q(1)}^{m_i}), \cdots, (v_{q(n)}^{m_i}, t_{q(n)}^{m_i})]$ where for any two elements k and j in the sequence, if $k < j$, then $t_{q(k)}^{m_i} < t_{q(j)}^{m_i}$, meaning that $v_{q(k)}^{m_i}$ spread the information earlier than $v_{q(j)}^{m_i}$ did. Therefore, given n nodes, the number of all possible diffusion networks are reduced to $n!$. In order to further alleviate the sparsity, we incorporate social proximity and social dimensions in Section 4.3.2.

However, a possible problem of temporally sequencing spreaders is the loss of dependencies between users. Given v_i^m and v_j^m where $e_{ij} \in E$. If $t_i^m < t_j^m$, it is likely that user i spreads it to j or j is influenced by i (Gomez Rodriguez *et al.*, 2010). Such direct dependency will be of vital importance in characterizing the information. For example, the information flow from the controller account to the botnet followers is a key signal in detecting crowdturfing(Gu *et al.*, 2008). But if there is a spreader (u_k^m, t_k^m) where $< t_i^m < t_j^m$, in the sequence, i and j will be separated. Therefore, it would be appealing if the model can take advantage of dependencies between separated and distant items in a sequence. To this end, we propose to apply Recurrent Neural Networks (RNNs).

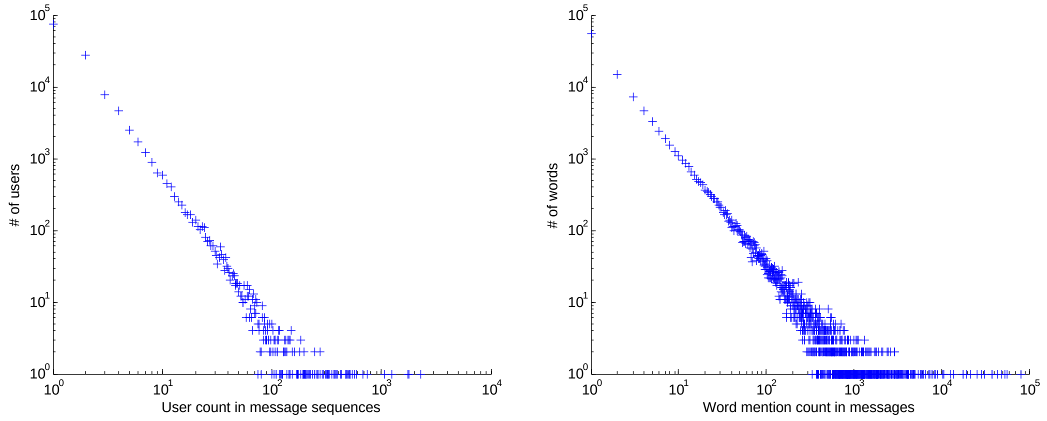
RNNs have been successfully applied in a myriad of domains for modeling sequential data (Goodfellow *et al.*, 2016), such as information retrieval (Palangi *et al.*, 2016), sentiment analysis (Socher *et al.*, 2013) and machine translation (Cho *et al.*, 2014). We propose to use an RNN to sequentially accept each spreader of a message and recurrently project it into a latent space with the contextual information from previous spreaders in the sequence. As the RNN reaches the end of the sequence, a prediction can be made based on the embedding vector produced by the hidden activations. In order to better encode the distant and separated dependencies, we further incorporate the Long Short-Term Memory cells into the RNN model, *i.e.,* the LSTM-RNN.

In information diffusion, the first spreader who initiates the diffusion process is more likely to be useful for classifying the message (Barbier *et al.*, 2013). Hence, we feed the spread sequence in the **reverse** order, where the first spreader in the sequence directly interacts with the prediction result, and thus it has more impact. Each spreader is represented by a local RNN. Parameters $\mathbf{W}$ of RNNs are shared across each replication in the sequence and h' is the previous recurrent output sent between RNNs to exploit the contextual information. In order to make the prediction, the last local RNNs are taking the first spreader's attribute vector, prior recurrent output (and the label of the message) as input to predict the category of the message (or to train the RNNs model). In this section, we set the hidden node size (k) as 10. The way we obtain the attribute vector of nodes is introduced in Section 4.3.2.

Having chosen LSTM-RNNs as our method to classify messages, we now need a suitable way of learning attribute vectors $\mathbf{f}$, for social media users. An intuitive way is to utilize the social network graph G to generate embedding vectors (Perozzi *et al.*, 2014; Tang *et al.*, 2015), and feed sequences of embedding vectors to the LSTM-RNNs (Palangi *et al.*, 2016). Such embedding-based preprocessing for sequential data has been widely used for natural language processing. We follow the practice since 1) several social graph embedding approaches have been proven useful for classification tasks, such as LINE (Tang *et al.*, 2015) and DeepWalk (Perozzi *et al.*, 2014), and 2) users appear in spread traces follow similar distribution of how words appear in the social media posts.

Figure 4.1 illustrates the distribution of users and words. The distribution in Figure 4.1(a) comes from a real-world Twitter message trace dataset showing how users appear in message traces. The distribution in Figure 4.1(b) comes from the same dataset showing how words appear in message content. They both follow a power-law distribution, which motivates us to embed users into low dimensional vectors, as how embedding vectors of words are used in natural language processing (Kim, 2014; Palangi *et al.*, 2016). Several graph embedding algorithms are available, we will compare their performance and provide our solution and reasons behind our choice in the next subsection. For the rest of the subsection, we will introduce the optimization for the proposed LSTM-RNNs.

We show the training of the proposed LSTM-RNNs in Algorithm 6. We input the labeled spreader sequences X and the corresponding labels Y, which are randomly split into a training and a validation set in line *2*. In addition to the maximum number of iterations Max_{iter}, we also have a function $EarlyStop()$ for controlling early termination of the training, which takes the loss on the validation set as the input. In line *1*, we initialize the model parameters randomly with Gaussian distribution. From line *3* to *7*, we update $\mathbf{W}$ with training data until the maximum epoch is reached or the early termination condition is met. The loss function used in line *4* is shown



(a) Frequency of Users in Social Media Message Traces.

(b) Frequency of Words in Social Media Message.

Figure 4.1: The Frequency of Users Appearing in Traces of Social Media Messages Follows a Power-law Distribution, Which Is Similar to the Distribution of Word Frequencies in Messages.

below:

$$\sum_{i=1}^{|X_{tr}|} |Y_{tr} = 0| y_i \log(\hat{y}_i) + |Y_{tr} = 1|(1 - y_i)(\log(1 - \hat{y}_i)), \tag{4.1}$$

where y_i is the true label of i and $\hat{y}_i$ is the corresponding prediction. So Eq.(4.1) calculates the cross entropy between the true labels and the prediction. $|Y_{tr} = 0|$ ($|Y_{tr} = 1|$) is the number of negative (positive) instances in the training set. Since we aim to work on multi-label classification, the data is naturally imbalanced when we model one of them, introducing the weight helps the model balance the gradient of skewed data. In next subsection, we will introduce how we generate embeddings and the reason behind our choice.

4.3.2 Embedding of Users

Given the framework of sequence modeling, the next problem is to find the proper embedding method that captures the intrinsic features of social media users. As discussed previously, using embedding vectors can help alleviate the data sparsity

Algorithm 2 Training Algorithm of LSTM-RNNs

```
Input: Labeled sequences and labels    $X, Y$
  Maximum number of iterations:    $Max_{iter}$
  Early termination function :    $EarlyStop()$
Output: Weights of LSTM-RNNs:    $\mathbf{W}$
  Initialize $\mathbf{W}$ randomly with Gaussian distribution, $VLoss[Max_{iter}]$, $i = 0$
  Split $X$ and $Y$ into training and validation set, $(X_{tr}, Y_{tr})$ and $(X_{val}, Y_{val})$
  do
      Train RNNs with $(X_{tr}, Y_{tr})$ for 1 epoch with Eq.(4.1)
      Test RNNs with $(X_{val}, Y_{val})$ to obtain loss $VLoss[i]$
      $i = i + 1$
  while $EarlyStop(VLoss, i) = FALSE$ AND $(i < Max_{iter})$
```

through leveraging social proximity and social dimensions. In this section, among the existing embedding methods, we will mainly focus on two state-of-the-art approaches that have been proven effective on social graphs, LINE (Tang *et al.*, 2015) and DeepWalk (Perozzi *et al.*, 2014). Both LINE and DeepWalk aim to provide a representation for data instances that captures the inherent properties, such as social proximity.

These methods mainly focus on the microscopic structure of networks. For example, first-order proximity constrains users that are connected to be similar and second-order proximity constrains users that have common friends to be similar. LINE achieves this by sampling such nodes from the network and updating their representations jointly, while DeepWalk samples a sequence of data with a random walk algorithm. Nevertheless, for a large social graph, some mesoscopic structure such as social dimensions (Tang and Liu, 2009) and community structures (Yang *et al.*, 2013)

Table 4.1: Average Euclidean Distance Between Nodes with Low Dimensional Representation.

Method	1^{st}-degree	2^{nd}-degree	Intra-group
LINE	**5.16**	**5.00**	10.76
DeepWalk	7.74	7.69	6.04
SocDim	6.87	6.12	**4.55**

are more useful in characterizing information (Laumann and Pappi, 2013). Therefore, the ideal embedding method should be able to capture both local proximity and community structures.

Table 4.1 illustrates our results of using different embedding methods. We test LINE, DeepWalk and SocDim (Tang and Liu, 2009) on Twitter data and show the distance between neighbors with the new representation. We also detect community structures in the network and calculate the average of distances between nodes that are in the same community. The community detection algorithm is an accelerated version of Louvain method (Blondel *et al.*, 2008). As shown in the table, LINE captures the first and second-degree proximity, while SocDim best captures the community-wise proximity. Based on the random walk, DeepWalk achieves better community-wise proximity, however, it is still outperformed by SocDim, which directly models the community structure.

In order to capture both the social proximity and community-wise similarity among users, we propose a principled framework that directly models both kinds of information. Given the social graph G, we can derive an adjacency matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$, where n is the number of users. Our goal is to learn a transformation matrix $\mathbf{M} \in \mathbb{R}^{n \times k}$ which converts users to a latent space with the dimensionality of k. Note that we reuse k for brevity of presentation, and the number of features and hidden nodes in the LSTM-RNNs are not necessarily the same. In order to capture the community-wise similarity, we introduce two auxiliary matrices, a community indicator matrix $\mathbf{H} \in \mathbb{R}^{n \times g}$, where g is the number of communities and $tr(\mathbf{H}\mathbf{H}^T) = n$ (only one element is 1 in each row and all the others are 0), and a community representation matrix $\mathbf{C} \in \mathbb{R}^{g \times k}$, where each row $\mathbf{c}_i$ is an embedding vector describing the community. In order to capture the community structure, we embed the problem into an attributed community detection model (Yang *et al.*, 2013):

$$\begin{aligned} &\min_{\mathbf{M},\mathbf{H},\mathbf{C}} \sum_{i=1}^{n} ||\mathbf{s}_i\mathbf{M} - \mathbf{h}_i\mathbf{C}||_2^2 + \alpha||\mathbf{H} - \mathbf{M}\mathbf{C}^T||_F^2, \\ &s.t. \quad tr(\mathbf{H}\mathbf{H}^T) = n, \end{aligned} \tag{4.2}$$

where $\mathbf{s}_i\mathbf{M}$ is the embedding vector and we regularize it to be similar to the representation of its corresponding community $\mathbf{h}_i\mathbf{C}$. The second term aims to achieve the intra-group coherence by predicting the community assignment by group the embedding vectors of users and communities (Yang *et al.*, 2013). The objective function in Eq.(5.1) aims to cluster nodes with embedding vectors. In order to further regularize the clusters to be social communities, we adopt a modularity maximization-based method, which has been widely used to detect communities with network information (Wu *et al.*, 2016a). Specifically, given the adjacency matrix $\mathbf{S}$ and the community membership indicator, the modularity is defined as follows (Tang and Liu, 2009):

$$Q = \frac{1}{2|E|} \sum_{i,j} (S_{ij} - \frac{d_i d_j}{2|E|})(\mathbf{h}_i\mathbf{h}_j^T), \tag{4.3}$$

where $|E|$ is the number of edges and d_i is the degree of i. $\mathbf{h}_i$ is the community assignment vector for i, and $\mathbf{h}_i\mathbf{h}_j^T = 1$ if i and j belong to the same community, otherwise $\mathbf{h}_i\mathbf{h}_j^T = 0$. $\frac{d_i d_j}{2|E|}$ is the expected number of edges between i and j if edges are placed at random. Modularity Q measures the difference between the number of actual edges within a community and the expected number of edges placed at random. An optimal community structure $\mathbf{H}$ should maximize the modularity Q. By defining the modularity matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$ where $B_{ij} = S_{ij} - \frac{d_i d_j}{2|E|}$ and suppressing the constant which has no effect on the modularity, we rewrite Eq.(4.3) as follows:

$$Q = tr(\mathbf{H}^T\mathbf{B}\mathbf{H}).$$

In order to guarantee that the embedding vectors preserve the community structure in the latent space, we propose to integrate modularity maximization into the embedding method. The objective function can be rewritten with the modularity maximization regularizer as follows:

$$\begin{aligned} &\min_{\mathbf{M},\mathbf{H},\mathbf{C}} \sum_{i=1}^{n} ||\mathbf{s}_i\mathbf{M} - \mathbf{h}_i\mathbf{C}||_2^2 + \alpha||\mathbf{H} - \mathbf{M}\mathbf{C}^T||_F^2 - \beta tr(\mathbf{H}^T\mathbf{B}\mathbf{H}) \\ &s.t. \quad tr(\mathbf{H}\mathbf{H}^T) = n, \end{aligned} \tag{4.4}$$

where β controls the influence of community structures. As discussed previously, the microscopic structure is also of vital importance for generating embedding vectors. In order to jointly consider both mesoscopic and microscopic structures, we decompose $\mathbf{M}$ into a conjunction of a global model parameter $\tilde{\mathbf{M}}$ and a localized variable $\mathbf{M_i}$ for each user i ($\mathbf{M} = \tilde{\mathbf{M}} + \mathbf{M_i}$ for each user i). Therefore, $\tilde{\mathbf{M}}$ captures the community structure and $\mathbf{M_i}$ can be used to directly apprehend the microscopic structure between nodes. Motivated by recent research on network regularization, we fortify the representation of nodes with proximity by the network lasso regularization term (Hallac *et al.*, 2015):

$$\sum_{i,j} \mathbf{A}_{ij}||\mathbf{M}_i - \mathbf{M}_j||_F^2,$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the microscopic structure matrix, $A_{ij} = 1$ if we aim to preserve the proximity between i and j in the latent space. Following conventional graph embedding practices (Tang *et al.*, 2015), we consider first- and second-degree proximity, meaning that $A_{ij} = 1$ if i and j are connected or share a common friend. Note that A can be specified with particular applications. Imposing the Frobenius norm of the difference between $\mathbf{M}_i$ and $\mathbf{M}_j$ incentivizes them to be the same when $A_{ij} = 1$. By incorporating the network lasso regularizer, the objective function can be reformulated as follows:

$$\begin{aligned}
&\min_{\mathbf{M},\mathbf{H},\mathbf{C}} \sum_{i=1}^{n} ||\mathbf{s}_i(\tilde{\mathbf{M}} + \mathbf{M_i}) - \mathbf{h}_i\mathbf{C}||_2^2 + \alpha||\mathbf{H} - \tilde{\mathbf{M}}\mathbf{C}^T||_F^2 \\
&- \beta tr(\mathbf{H^T B H}) + \gamma \sum_{i,j} \mathbf{A}_{ij}||\mathbf{M}_i - \mathbf{M}_j||_F^2, \qquad (4.5)\\
&s.t. \quad tr(\mathbf{H H^T}) = n,
\end{aligned}$$

where γ controls the influence of the network lasso. As we can see, we establish the consensus relationship between mesoscopic and microscopic network structures by jointly considering the social communities and proximity. By introducing the global parameter $\tilde{\mathbf{M}}$ and the personal variable $\mathbf{M}_i$, we force both kinds of information to be preserved in the newly-learnt embedding vectors. However, Eq.(4.5) is not jointly convex to all the parameters $\mathbf{M}$,$\mathbf{H}$ and $\mathbf{C}$. In order to solve the problem, we separate the optimization into four subproblems and iteratively optimize them. We will introduce details of the optimization for the rest of the section.

Update $\tilde{\mathbf{M}}$ while fixing $\mathbf{M_i}$, $\mathbf{H}$ and $\mathbf{C}$: By removing terms that are irrelevant to $\tilde{\mathbf{M}}$, we obtain the following optimization problem:

$$\min_{\tilde{\mathbf{M}}} \sum_{i=1}^{n} ||\mathbf{s}_i\tilde{\mathbf{M}} + \mathbf{s}_i\mathbf{M_i} - \mathbf{h}_i\mathbf{C}||_2^2 + \alpha||\mathbf{H} - \tilde{\mathbf{M}}\mathbf{C}^T||_F^2, \qquad (4.6)$$

which is convex *w.r.t.* $\tilde{\mathbf{M}}$. In real applications, the number of users n may be huge. Hence, we adopt a gradient-based update rule as follows:

$$\tilde{\mathbf{M}} = \tilde{\mathbf{M}} - \tau\frac{\partial \epsilon_{\tilde{\mathbf{M}}}}{\partial \tilde{\mathbf{M}}}, \qquad (4.7)$$

where τ is the step size that can be obtained through backtracking line search (Nocedal and Wright, 2006). The derivative of $\tilde{\mathbf{M}}$ is shown as follows:

$$\frac{\partial \epsilon_{\tilde{\mathbf{M}}}}{\partial \tilde{\mathbf{M}}} = \mathbf{s}_i^T \sum_{i=1}^{n}(\mathbf{s}_i\tilde{\mathbf{M}} + \mathbf{s}_i\mathbf{M_i} - \mathbf{h}_i\mathbf{C}) + \alpha(\mathbf{H} - \tilde{\mathbf{M}}\mathbf{C}^T)\mathbf{C}. \qquad (4.8)$$

Update $\mathbf{M_i}$ while fixing $\tilde{\mathbf{M}}$, $\mathbf{H}$ and $\mathbf{C}$: By removing terms that are irrelevant to $\mathbf{M}_i$, we obtain the following optimization problem:

$$\min_{\mathbf{M}_i} \sum_{i=1}^{n} ||\mathbf{s}_i\tilde{\mathbf{M}} + \mathbf{s}_i\mathbf{M_i} - \mathbf{h}_i\mathbf{C}||_2^2 + \gamma \sum_{i,j} A_{ij}||\mathbf{M}_i - \mathbf{M}_j||_F^2, \tag{4.9}$$

which is convex *w.r.t.* $\mathbf{M}_i$. Similarly, we derive the gradient:

$$\frac{\partial \epsilon_{\mathbf{M}_i}}{\partial \mathbf{M}_i} = \mathbf{s}_i^T \sum_{i=1}^{n} (\mathbf{s}_i\tilde{\mathbf{M}} + \mathbf{s}_i\mathbf{M_i} - \mathbf{h}_i\mathbf{C}) + \gamma \sum_{i,j} A_{ij}(\mathbf{M}_i - \mathbf{M}_j). \tag{4.10}$$

Update $\mathbf{C}$ while fixing $\tilde{\mathbf{M}}$, $\mathbf{M_i}$, and $\mathbf{H}$: By removing terms that are irrelevant to $\mathbf{C}$, we obtain the following optimization problem:

$$\min_{\mathbf{C}} \sum_{i=1}^{n} ||\mathbf{s}_i(\tilde{\mathbf{M}} + \mathbf{M_i}) - \mathbf{h}_i\mathbf{C}||_2^2 + \alpha||\mathbf{H} - \tilde{\mathbf{M}}\mathbf{C}^T||_F^2, \tag{4.11}$$

which is convex *w.r.t.* $\mathbf{C}$. Similarly, the gradient can be obtained as:

$$\frac{\partial \epsilon_{\mathbf{C}}}{\partial \mathbf{C}} = \sum_{i=1}^{n} \mathbf{h}_i^T(\mathbf{h}_i\mathbf{C} - \mathbf{s}_i\tilde{\mathbf{M}} - \mathbf{s}_i\mathbf{M_i}) + \alpha(\tilde{\mathbf{M}}\mathbf{C}^T - \mathbf{H})^T\tilde{\mathbf{M}}. \tag{4.12}$$

Update $\mathbf{H}$ while fixing $\tilde{\mathbf{M}}$, $\mathbf{M_i}$, and $\mathbf{C}$: By removing terms that are irrelevant to $\mathbf{H}$, we obtain the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{H}} ||\mathbf{SM} - \mathbf{HC}||_F^2 + \alpha||\mathbf{H} - \tilde{\mathbf{M}}\mathbf{C}^T||_F^2 - \beta tr(\mathbf{H^T}(\mathbf{S} - \hat{\mathbf{B}})\mathbf{H}), \\ &s.t. \quad tr(\mathbf{H}\mathbf{H}^T) = n, \end{aligned} \tag{4.13}$$

where $\hat{\mathbf{B}}_{ij} = \frac{d_i d_j}{2|E|}$. Consider that $\mathbf{H}$ is an indicator matrix, the constraint makes the problem in Eq.(4.13) NP-complete, which is extremely difficult to solve. In order to cope with the problem, we relax the constraint to orthogonality $\mathbf{H}^T\mathbf{H} = \mathbf{I}$ and nonnegativity $\mathbf{H} \geq 0$ and reformulate the objective function as follows:

$$\begin{aligned} \epsilon_{\mathbf{H}} = &- \beta tr(\mathbf{H}^T\mathbf{S}\mathbf{H}) + \beta tr(\mathbf{H}^T\hat{\mathbf{B}}\mathbf{H}) \\ &+ ||\mathbf{SM} - \mathbf{HC}||_F^2 + \alpha||\mathbf{H} - \hat{\mathbf{M}}\mathbf{C}^T||_F^2 \\ &+ \lambda||\mathbf{H}^T\mathbf{H} - \mathbf{I}||_F^2, \end{aligned} \tag{4.14}$$

Table 4.2: Statistics of the Datasest Used in This Study.

	Messages	Posts	Unique Users	Class Ratio
Real News	68,892	288,591	121,211	0.27(b):0.25(t):0.37(e):0.11(m)
Fake News	3,600	17,613	9,153	0.5:0.5

where $\lambda > 0$ should be a large number to guarantee the orthogonal constraint to be satisfied, and we set it as 10^8 in this section. We then utilize the property that $||\mathbf{X}||_F^2 = tr(\mathbf{X}^T\mathbf{X})$ to reformulate the loss function as follows:

$$\begin{aligned} \epsilon_{\mathbf{H}} = & - & \beta tr(\mathbf{H}^T\mathbf{S}\mathbf{H}) + \beta tr(\mathbf{H}^T\hat{\mathbf{B}}\mathbf{H}) \\ & + & tr(\mathbf{S}\mathbf{M}\mathbf{M}^T\mathbf{S}^T + \mathbf{H}\mathbf{C}\mathbf{C}^T\mathbf{H}^T - 2\mathbf{S}\mathbf{M}\mathbf{C}^T\mathbf{H}^T) \\ & + & \alpha tr(\mathbf{H}\mathbf{H}^T + \hat{\mathbf{M}}\mathbf{C}^T\mathbf{C}\tilde{\mathbf{M}}^T - 2\mathbf{H}\mathbf{C}\tilde{\mathbf{M}}^T) \\ & + & \lambda tr(\mathbf{H}^T\mathbf{H}\mathbf{H}^T\mathbf{H} - 2\mathbf{H}^T\mathbf{H} + \mathbf{I}) + tr(\mathbf{\Theta}\mathbf{H}^T), \end{aligned} \tag{4.15}$$

where $\mathbf{\Theta} = [\Theta_{ij}]$ is a Lagrange multiplier matrix to impose the nonnegative constraint. Set the derivative of $\frac{\partial \epsilon_{\mathbf{H}}}{\partial \mathbf{H}}$ to 0, we have:

$$\begin{aligned} \mathbf{\Theta} & = & 2\mathbf{S}\mathbf{H} - 2\beta\tilde{\mathbf{B}}\mathbf{H} - 2\mathbf{C}\mathbf{C}^T\mathbf{H}^T + 2\mathbf{S}\mathbf{M}\mathbf{C}^T \\ & - & 2\alpha\mathbf{H}^T + 2\alpha\mathbf{C}\tilde{\mathbf{M}}^T - 4\lambda\mathbf{H}\mathbf{H}^T\mathbf{H} + 4\lambda\mathbf{H}. \end{aligned} \tag{4.16}$$

Following the Karush-Kuhn-Tucker (KKT) condition for the nonnegativity, we have the equation as follows:

$$\begin{aligned} & (2\mathbf{S}\mathbf{H} - 2\beta\tilde{\mathbf{B}}\mathbf{H} - 2\mathbf{C}\mathbf{C}^T\mathbf{H}^T + 2\mathbf{S}\mathbf{M}\mathbf{C}^T - 2\alpha\mathbf{H}^T \\ & +2\alpha\mathbf{C}\tilde{\mathbf{M}}^T - 4\lambda\mathbf{H}\mathbf{H}^T\mathbf{H} + 4\lambda\mathbf{H})_{ij}H_{ij} = \theta_{ij}H_{ij} = 0, \end{aligned} \tag{4.17}$$

which is the fixed point equation that the solution must satisfy at convergence. The update rule for $\mathbf{H}$ can be written as follows:

$$\mathbf{H} = \mathbf{H} \odot \sqrt{\frac{-2\beta\tilde{\mathbf{B}}\mathbf{H} + \sqrt{\Delta}}{8\lambda\mathbf{H}\mathbf{H}^T\mathbf{H}}}, \tag{4.18}$$

where Δ is defined as:

$$\begin{aligned}\Delta &= 2\beta(\tilde{\mathbf{B}}\mathbf{H}) \odot (\tilde{\mathbf{B}}\mathbf{H}) + 16\lambda(\mathbf{H}\mathbf{H}^T\mathbf{H}) \\ &\odot (2\mathbf{S}\mathbf{H} - 2\mathbf{C}\mathbf{C}^T\mathbf{H}^T + 2\mathbf{S}\mathbf{M}\mathbf{C}^T \\ &- 2\alpha\mathbf{H}^T + 2\alpha\mathbf{C}\tilde{\mathbf{M}}^T + 4\lambda\mathbf{H}).\end{aligned} \tag{4.19}$$

The convergence of Eq.(4.19) can be proven as an instance of nonnegative matrix factorization (NMF) problem (Lee and Seung, 2001).

4.3.3 Time Complexity

TraceMiner consists of two components, LSTM-RNNs and the embedding method. Though LSTM-RNNs take $O(|E|+|V|)$-time for backpropagations, the scalability can be easily increased with deep learning software library like Theano[2], especially when GPU is available.

Since the number of users is usually far larger than the number of features and number of communities, the embedding method takes $O(n^2)$-time. Only matrix multiplication is used in all update rules, so the optimization can be accelerated by utilizing matrix optimization library like OpenBLAS[3].

4.4 Experiments

In this section, we introduce experiment details to validate the effectiveness of the proposed framework. Through the experiments, we aim to answer two questions:

- How well can network information be used to classify social messages compared with content information?

[2] http://deeplearning.net/software/theano/

[3] http://www.openblas.net/

- How effective are the LSTM-RNNs by integrating with the proposed embedding method?

Therefore, we test the methods on two different classification tasks with real-world datasets and include both content-based and network-based baselines for comparison.

4.4.1 Datasets

Over 200 million posts are posted per day on Twitter[4] and the popularity has made Twitter a testbed for information filtering research. In this section, we aim to collect a large dataset that includes tweets about specific messages. Following (Qazvinian *et al.*, 2011), we leverage Twitter Search API[5] to retrieve tweets of interests by compiling queries with certain topics.

We deal with two tasks in this section, standard news classification and fake news detection. News classification is a classical multi-label text categorization problem and existing efforts have mainly focused on the content. We obtain a news dataset which was originally used for content-based classification[6] by selecting news that has at least two posts on Twitter. Queries for Twitter Search API are compiled by words in the title of the corresponding news. Based on the spreaders of news, we try to use TraceMiner to classify the news into four categories: business (b), science and technology (t), entertainment (e), medical (m). Statistics about the dataset are shown in Table 7.2. We sample $68,892$ pieces of news, which relate to $288,591$ posts with $121,211$ unique users. The ratio of different categories is also presented.

The other task is fake news detection. The openness of social media platforms enables timely information to be spread at a high rate. Meanwhile, it also allows

[4] https://blog.twitter.com/2011/200-million-tweets-per-day

[5] https://dev.twitter.com/rest/public/search

[6] https://archive.ics.uci.edu/ml/datasets/News+Aggregator

for the rapid creation and dissemination of fake news. Following (Qazvinian *et al.*, 2011), we retrieve tweets related to fake news by compiling queries with a fact-checking website. In this section, we choose Snopes[7] to obtain ground truth, where we collect articles tagged with fake news[8]. In order to obtain non-fake news posts pertaining to the same topic, we extract keywords in regular expressions as queries to retrieve posts. Statistics of the dataset is shown in Table 7.2. We collect $3,600$ messages with 50% are fake news.

Table 4.3: The F_1-measure of Different Methods on the Task of Social Media News Categorization.

	Training Ratio	10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F_1(%)	SVM	0.6967	0.7138	0.7447	0.7577	0.7988	0.8096	0.8499	0.8787	0.8996
	XGBoost	0.7121	0.7349	0.7512	0.7794	0.8248	0.8250	0.8638	0.8951	0.9047
	TM(DeepWalk)	0.7895	0.8081	0.8149	0.8374	0.8569	0.8627	0.8852	0.8917	0.9184
	TM(LINE)	0.7691	0.7926	0.8163	0.8379	0.8467	0.8744	0.8980	0.9106	0.9253
	TraceMiner	**0.8275**	**0.8460**	**0.8658**	**0.8835**	**0.8885**	**0.9141**	**0.9218**	**0.9357**	**0.9380**
Macro-F_1(%)	SVM	0.6988	0.7260	0.7425	0.7754	0.7665	0.7872	0.8118	0.8314	0.8722
	XGBoost	0.7305	0.7438	0.7857	0.7887	0.8144	0.8344	0.8726	**0.8941**	0.9044
	TM(DeepWalk)	0.7746	0.8010	0.8156	0.8313	0.8377	0.8611	0.8646	0.8734	0.8839
	TM(LINE)	0.7561	0.7895	0.8019	0.8138	0.8235	0.8568	0.8775	0.8896	**0.9153**
	TraceMiner	**0.8181**	**0.8347**	**0.8359**	**0.8549**	**0.8635**	**0.8788**	**0.8779**	0.8882	0.9064

4.4.2 Experimental Settings

A core contribution of our work is the idea that spreaders of information can be used to predict message categories. Therefore, we try to test the effectiveness of the proposed method comparing with the state-of-the-art content-based approaches. We experiment a variety of approaches, and report the following two which achieve better results.

[7] http://www.snopes.com/

[8] https://www.snopes.com/tag/fake-news/

- **SVM** (Joachims, 1998) trains on content information, which is first preprocessed with Stanford CoreNLP toolkit (Manning *et al.*, 2014). We adopt bigram and trigram features based on results on the validation set.

- **XGBoost** (Chen and Guestrin, 2016) is an optimized distributed gradient boosting library that implements machine learning algorithms under the Gradient Boosting framework. It has been successfully applied to various problems and competitions. We feed it with the preprocessed content produced by Stanford CoreNLP. XGBoost presents the best results among all content-based algorithms we tested.

We propose a novel embedding method to cater to TraceMiner. In order to evaluate its effectiveness, we introduce two variants of TraceMiner and present their results for comparison:

- **TM(DeepWalk)** is a variant of TraceMiner by adopting the embedding vectors from DeepWalk as input. As discussed earlier, DeepWalk captures proximity between nodes with random walk: nodes that are sampled together with one random walk are forced to preserve the similarity in the latent space. Therefore, DeepWalk does not directly model the first and second-degree proximity or the community structure.

- **TM(LINE)** is a variant of TraceMiner by adopting the embedding vectors from LINE. LINE models first and second-degree proximity while does not consider the community structure between users.

To test the prediction accuracy in terms of both precision and recall, we adopted the F_1-measure to evaluate the performance. Since there are multiple labels to be predicted, for each task t, F_1^t can be computed. In order to get the overall performance, we first adopt the Macro-averaged F_1-measure as:

$$Macro - F_1 = \frac{1}{|\boldsymbol{T}|} \sum_{t \in \boldsymbol{T}} F_1^t, \tag{4.20}$$

where $\boldsymbol{T}$ is the set of all identity labels and F_1^t is the F_1-measure of task t.

A possible problem of Macro-F_1 is, since the sizes of different categories are different, the task with fewer instances may be overemphasized. In order to cope with this problem, we adopted Micro-averaged F_1-measure. First, we calculate the micro averaged precision and recall:

$$\begin{aligned} Micro - precision &= \frac{\#TP}{\#TP + \#FP} \\ Micro - recall &= \frac{\#TP}{\#TP + \#FN}, \end{aligned} \tag{4.21}$$

where #TP is the number of true positives, #FP is the number of false positives and #FN is the number of false negatives. Micro-F_1 is the harmonic average of Micro-precision and Micro-recall.

4.4.3 Experimental Results

Table 4.4: The F_1-measure of Different Methods on the Task of Fake News Detection.

Training Ratio	10%	20%	30%	40%	50%	60%	70%	80%	90%
SVM	0.5825	0.5779	0.6122	0.6194	0.6658	0.7114	0.7224	0.7252	0.7581
XGBoost	0.6558	0.7004	0.7002	0.7153	0.7288	0.7703	0.7984	0.8115	0.8226
TM(DeepWalk)	0.7804	0.7810	0.8078	0.8264	0.8194	0.8491	0.8542	0.8738	0.8894
TM(LINE)	0.7542	0.7547	0.7913	0.8015	0.8083	0.8485	0.8733	0.8936	0.8971
TraceMiner	**0.7867**	**0.7935**	**0.8344**	**0.8459**	**0.8547**	**0.8751**	**0.8988**	**0.9089**	**0.9124**

Social Media News Categorization: The performance of different methods on Twitter News data with varying training ratio, from 10% to 90%, is illustrated in Table 4.3. For each experiment, samples are randomly split into training and testing set. We repeat this process 10 times and report the average results. The highest performance under each setting is highlighted in bold face.

In terms of Micro-F_1, our proposed model TraceMiner outperforms all the baselines and its variations, TM(DeepWalk), TM(LINE). Diffusion-based methods perform better than content-based methods. XGBoost performs slightly better than SVM. TM(DeepWalk) is the runner-up method for 10%, 20% and 50%, and TM(LINE) is the runner-up for the rest cases. The result shows that when less network data is available, the random walk-based approach produces better embeddings of users; And a more deterministic method constraining on social proximity better apprehends user behaviors when the network information is more complete. TraceMiner achieves the best result for all tasks. By jointly modeling the microscopic and mesoscopic structures, TraceMiner is more robust to data sparsity.

In terms of Macro-F_1, XGBoost outperforms SVM for all cases. Similar pattern has again been observed: TM(DeepWalk) outperforms TM(LINE) with less training information, while TM(LINE) outperforms TM(DeepWalk) when the information is more complete. TraceMiner still performs the best among most cases until we increase the training ratio up to 80%. XGBoost and TM(LINE) achieves the best result for 80% and 90%, respectively. Two observations can be made here: with more training information becoming available, 1) the margin between proposed methods and the content-based methods becomes smaller; and 2) the margin between TraceMiner and its variants TM(LINE) and TM(DeepWalk) becomes smaller. Based on the observations we can draw conclusions that TraceMiner is more useful when less training information is available, and the proposed TraceMiner can well handle scarce data in the early phase of learning when less training information is known. XGBoost gets the best when 80% of information is available. Since text-based categorization is a well-studied problem, and it is easy to solve when rich information is available, TraceMiner will be able to complement those cases that are difficult for content-based approaches to deal with, and such cases are pervasively present in social media mining tasks where content information is insufficient and noisy.

Another observation that again validates our findings is that TraceMiner performs better in terms of Micro-F_1. As shown in Eq.(4.20) and (4.21), in a multi-label classification task, the category with fewer instances is more advantageous for Macro-F_1. The results show that TraceMiner actually ends up with correctly classifying more instances.

Fake News Detection: The performance of different methods on Twitter fake news data with varying training ratio, from 10% to 90%, is illustrated in Table 4.4. Since the dataset is balanced, Micro- and Macro-F_1 are the same, so only one set of results are presented. For the content-based approaches, XGBoost consistently outperforms SVM for all cases. For the two variants of TraceMiner, similar patterns are observed: TM(DeepWalk) outperforms TM(LINE) when less training information is available. TM(LINE) outperforms TM(DeepWalk) when more information is available for training. It again proves that random walk-based sampling is more effective for scarce data, and proximity-based regularization better captures data structures with more training information.

An interesting difference between the results for fake news and the previous experiment is the larger margin between proposed methods and content-based methods. Unlike posts related to news where the content information is more self-explanatory, content of posts about fake news is less descriptive. Intentional spreaders of fake news may manipulate the content to make it look more similar to non-rumor information. Hence, TraceMiner can be useful for many emerging tasks in social media where adversarial attacks are present, such as detecting rumors and crowdturfing. The margin between content-based approaches and TraceMiner becomes smaller when more information is available for training, however, in these emerging tasks, training information is usually time-consuming and labor-intensive to obtain.

Another point we would like to discuss is the performance when the training information is very insufficient. When 10% of information is available, SVM has an F_1 score of 58% which is slightly better than a random guess, while TraceMiner has an F_1 score of 78%. Although such margin is reduced when more information is available, the optimal performance with very few training information is of crucial significance for tasks which emphasize on the earliness. For example, detecting fake news at an early stage is way more meaningful than detecting it when 90% percent of its information is known (Sampson *et al.*, 2016b; Qazvinian *et al.*, 2011; Wu *et al.*, 2017b). In conclusion, TraceMiner provides an effective method for modeling messages diffused in social media with only network information, which provides a complementary tool for emerging tasks that require earliness and/or suffers from the scarcity of content information.

4.5 Summary

In this section, we aim to classify messages spread in social networks, which is a fundamental problem for social media mining. We observe that for many emerging tasks, content information is usually insufficient or less descriptive, while pervasively available network information is left unused. Therefore, we propose a novel method TraceMiner that classifies social media messages with diffusion traces in social networks. To address the problem, we propose an end-to-end classification model based on LSTM-RNNs. In order to alleviate the data sparsity, we propose an embedding method that captures both social proximity and community structures. Experimental results with real-world datasets show that TraceMiner effectively classifies social media messages and is especially useful when content information is insufficient.

Chapter 5

MISINFORMATION DETECTION AT AN EARLY STAGE

In this chapter, we focus on delivering an effective misinformation detection model in an early stage. Since misinformation evolves and spreads rapidly in social networks, ignoring earliness of intervention makes the intervening campaign downgrade fast due to the evolved content. I will introduce the computational challenge of data scarcity at an early stage. Then I will formally define the computational problem, and present the proposed method. We conduct experiments to evaluate the effectiveness and earliness of the proposed method against the state-of-the-art approaches.

5.1 Data Scarcity at an Early Stage

The prevalence of social media has revolutionized the way of information dissemination and communication. The openness of social media platforms enables timely information to be spread at a high rate. Meanwhile, it also allows for the rapid creation and dissemination of rumors, which could cause catastrophic effects in the real world within a short period. For example, on April 23rd 2013, the hacked Twitter account of Associate Press posted a false claim of an attack on the White House, which was soon covered by news agencies, and wiped out $136 billion in the stock market within two minutes[1]. It would be appealing if emerging rumors could be automatically detected in its early stage.

Classical rumor detection methods highly depend on learning patterns from manually labeled data. A straightforward way is to learn a classifier or regressor based on

[1] http://www.bloomberg.com/news/articles/2013-04-23/dow-jones-drops-recovers-after-false-report-on-ap-twitter-page

labeled rumors, and then the built model can be employed to determine the credibility of a new message or user. However, in real-world applications, annotating a rumor dataset could be time-consuming and labor-intensive, sometimes even impractical. The labeling bottleneck brings in an unavoidable delay for existing systems, resulting in significant challenges to enable the system to detect new rumors in a timely manner. Therefore, it would be desirable to develop a way for rumor detection without the labeling process.

While the problem of detecting rumors on social media is relatively new, rumors have been extensively investigated for years in social and psychological studies. The literature can be traced back to (Allport and Postman, 1947). A conventional methodology of studying rumors is to analyze the testimonies. The origins, consequences and potential impact of a rumor can be well estimated by linking it to a historical rumor through examining the behaviors of social participants who are exposed to it (Anthony, 1973; Rosnow, 1991), since *similar* rumors usually trigger similar reactions, such as curiosity, inquiry, and anxiety. Although the content on social networks is informal, its significant role in understanding a rumor has been found (Oh *et al.*, 2013). Motivated by the previous findings, we explore the possibility of using the abundant labeled data from prior rumors to facilitate the detection of an emerging rumor.

However, it is particularly difficult and challenging to directly use labeled data from one rumor to build a detection model for the other, *a.k.a.* cross-training. Cross-training can be successfully applied to problems of which different tasks are similar. Since rumor data is highly topic-sensitive, the vocabulary and word choice may vary substantially between different rumors. Therefore, directly applying an existing dataset would lead to the inclusion of noisy features and thus may negatively inhibit the prediction accuracy. In addition, since a certain category of rumors may trigger specific reactions, *e.g.,* wedge-driving rumors cause hatred and atrocity rumors arouse astonishment, it is ideal to find useful patterns within a category. Due to the lack of availability of the category information, it is difficult to find the scarce patterns out of miscellaneous labeled data. Also, since social media users tend to communicate concisely and casually (Kietzmann *et al.*, 2011), the short content may further exacerbate the scarcity problem.

In order to tackle the aforementioned challenges, we present a novel learning framework to detect emerging rumors with existing labeled data from prior rumors. The proposed framework is built upon a sparse representation model, and it jointly selects descriptive features from prior labeled data and trains the topic-independent classifier with selected features. The proposed framework extends the earliness bottleneck of current rumor detection methods.

5.2 Problem Statement

$\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m\} \in \mathbb{R}^{m \times n}$ is the data matrix with each row $\mathbf{d}_i \in \mathbb{R}^n$ being a data instance and each column $\mathbf{f}_i \in \mathbb{R}^m$ being a vector of each feature. $\mathbf{y} \in \{-1, 1\}^m$ is the label vector for training data. $\mathbf{y}_i = 1$ if i refers to a rumor, and otherwise, $\mathbf{y}_i = -1$. Given the data matrix $\mathbf{D}$, label vector $\mathbf{y}$, we aim to learn a predictor that accurately classifies rumors and non-rumors based on the social media posts.

5.3 Early Detection with Prior Label Information

5.3.1 Motivation

In Table 5.1, we display rumors about two topics. The first rumor in Table 5.1(a) is about endorsements for the presidential candidate. The rumor says a famous evangelist urged Christians to vote for Donald Trump, otherwise they will face death

Table 5.1: Two Real-world Examples of Social Media Rumors.

(a) An Example Rumor about the Presidential Election and the Corresponding Social Media Posts.

Rumor	Rightwing Christian says elect #trump or face #deathcamps run by #liberals http://bit.ly/2as5MJ5 .
Post #1	Christian conservative gets political. Can't fix stupid but it can be blocked.
Post #2	So, when did bearing false witness become a Christian value?
Post #3	Graham Says Christians Must Support Trump or Face Death Camps. Does he still claim to be a Christian?

(b) An Example Rumor about the Ferguson Protests and the Corresponding Social Media Posts.

Rumor	A Ferguson protesting sign reads 'No Mother Should Fear for Her Son's Life Every Time He Robs a Store.'
Post #1	i've just seen the sign on fb. you can't fix stupid.
Post #2	THIS IS PURE INSANITY.. HOW ABOUT THIS STATEMENT.
Post #3	No Mother Should Have To Fear For Her Son's Life Every Time He Robs A Store #AllLivesMatt

camps. The following three sentences are posts of the rumor. The second rumor in Table 5.1(b) is about a Ferguson protester. The rumor says the sign that the protesters are holding reads "No Mother Should Fear for Her Son's Life Every Time He Robs a Store". The bias of word choice of different rumor topics makes it difficult for cross-training. For example, the classifiers trained on the first rumor, which use features such as "political" and "Christian" would be useless in identifying posts of the second rumor.

In the literature of social and psychological studies, both rumors can be categorized as wedge-driving rumors (Allport and Postman, 1947) that feed on hate. In the user posts, we find contents that express hostility similarly, such as "fix stupid" and "pure insanity". These similar expressions are useful in identifying future wedge-driving rumors, which may or may not be related to the two topics. Therefore, we aim to discover the topic-independent patterns in user posts.

5.3.2 Working of the Framework

In order to build the framework that can exploit prior labeled data, two main issues remain to be solved. An ideal case for selecting topic-independent features is that we group rumors by their categories and find discriminative features for each category, such as hatred features for wedge-driving rumors, worrying features for anxiety-arising rumors, and astonishment features for atrocity rumors. However, rumor categories are unavailable. In order to solve the problem, we adopt structure learning-based feature selection in this section. Motivated by recent research on unsupervised feature selection (Li *et al.*, 2016), for an unlabeled dataset, we can effectively select features by preserving the intrinsic structure of data. In our work, the structure is the rumor category, and within the same category, rumors trigger similar contents.

As conventional practices in unsupervised feature selection approaches, the selected features can then be used for training a classifier. However, the supervised information, *i.e.,* the rumor labels, has not been considered in the feature selection process, which leads to the issue that the selected features may fail to capture the key knowledge of rumors. A more coherent method is to integrate the feature selection and classification processes into a unified framework.

Figure 5.1 illustrates the three components of the proposed framework. The framework is built upon sparse representation learning methods, which simultaneously infers the category structure of rumor data and selects discriminative features. The rumor label is also jointly utilized by supervising the feature selection process which results in an optimal rumor classifier.

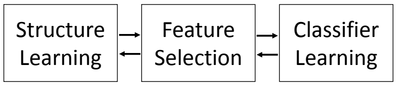


Figure 5.1: An Illustration of the Learning Procedure of the Proposed Framework. The Framework Consists of Three Components: Inferring Rumor Categories (Structure Learning), Selecting Discriminative Features, and Learning the Rumor Classifier.

5.3.3 Problem Definition

$\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m\} \in \mathbb{R}^{m\times n}$ is the data matrix with each row $\mathbf{d}_i \in \mathbb{R}^n$ being a data instance and each column $\mathbf{f}_i \in \mathbb{R}^m$ being a vector of each feature. $\mathbf{y} \in \{-1, 1\}^m$ is the label vector for training data. $\mathbf{y}_i = 1$ if i refers to a rumor, and otherwise, $\mathbf{y}_i = -1$. Given the data matrix $\mathbf{D}$, label vector $\mathbf{y}$, we aim to learn a predictor that accurately classifies rumors and non-rumors based on the social media posts.

Motivated by recent research on feature selection (Li *et al.*, 2016), we start with a matrix factorization formulation:

$$\min_{\mathbf{U},\mathbf{V}} \frac{1}{2}||\mathbf{D} - \mathbf{U}\mathbf{V}^T||_F^2,$$

where $||\cdot||_F$ denotes the *Frobenius* norm. The original data matrix is decomposed into two factors, $\mathbf{U} \in \mathbb{R}^{m\times k}$ is the low-rank representation of users, and $\mathbf{V} \in \mathbb{R}^{n\times k}$ is the low-rank representation of features with $k \ll n$. The factorization separates data from feature by k latent factors, which enables the clustering and feature selection to be jointly performed. In order to force the user factor $\mathbf{U}$ to be cluster indicators instead of latent factors, we impose a constraint on $\mathbf{U}$:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V}} \frac{1}{2}||\mathbf{D} - \mathbf{U}\mathbf{V}^T||_F^2, \\ &s.t. \quad \mathbf{U} \in \{0,1\}^{m\times k}, \mathbf{U}\mathbf{1} = \mathbf{1} \end{aligned} \tag{5.1}$$

where $\mathbf{1}$ is a vector with all elements equal to 1. The m rows are then clustered into k clusters. However, due to the constraint on $\mathbf{U}$, it is difficult to solve the problem in Eq.(5.1). Motivated by research on spectral clustering (Von Luxburg, 2007), we introduce an orthogonal constraint on the rows to relax it. Eq.(5.1) can then be rewritten as follows:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V}} \frac{1}{2}||\mathbf{D} - \mathbf{U}\mathbf{V}^T||_F^2, \\ &s.t. \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}, \mathbf{U} \geq \mathbf{0} \end{aligned} \tag{5.2}$$

where $\mathbf{I}$ is an identity matrix and thus rows in $\mathbf{U}$ are orthogonal to each other.

The orthogonal constraint ensures that data instances are clustered into different rumor categories. For each rumor category, we aim to select descriptive features. To this end, we try to select key features while force the unselected features to be zero. In the literature of sparse learning and feature selection, it can be done by imposing an $\ell_{2,1}$-norm (Hastie *et al.*, 2015). Motivated by recent studies on embedded feature selection (Wang *et al.*, 2015), we rewrite Eq.(5.2) as follows:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V}} \frac{1}{2}||\mathbf{D} - \mathbf{U}\mathbf{V}^T||_F^2 + \alpha||\mathbf{V}||_{2,1}, \\ &s.t. \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}, \mathbf{U} \geq \mathbf{0} \end{aligned} \tag{5.3}$$

where the $\ell_{2,1}$-norm regularizer selects features that best preserve the structure of clustering $\mathbf{U}$. α controls the extent of sparsity.

Through solving Eq.(5.3), we can obtain the low-rank representations. However, the labeled data that are available for distinguishing rumor and non-rumor content has not been exploited. The resultant representation would fail to capture the key signal that reveals the appearance of rumors of a category. Motivated by Collective Matrix Factorization-based relational learning (Singh and Gordon, 2008; Wu *et al.*, 2016a), we introduce a classification loss term in the objective function. We adopt the hinge loss used in Support Vector Machines (SVMs), and Eq.(5.3) is reformulated as:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V},\mathbf{w}} \frac{1}{2}||\mathbf{D} - \mathbf{U}\mathbf{V}^T||_F^2 + \alpha||\mathbf{V}||_{2,1} + \beta\sum_{i=1}^{m} h(\mathbf{u}_i\mathbf{V}^T\mathbf{w}\mathbf{y}_i), \\ &s.t. \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}, \mathbf{U} \geq \mathbf{0} \end{aligned} \tag{5.4}$$

where $h(\cdot)$ is the hinge loss and β controls the extent that the training information influences the feature selection and structure learning processes. $\mathbf{u}_i\mathbf{V}^T$ is the reconstructed formulation of a data instance. $\mathbf{w}$ is the model parameter of the SVMs, and $\mathbf{u}_i\mathbf{V}^T\mathbf{w}$ denotes the prediction with given low-rank representations. To make it convenient for optimization, we adopt the smoothed hinge loss (Rennie, 2005) for $h(\cdot)$ as follows:

$$h(\theta) = \begin{cases} \frac{1}{2} - \theta & \theta \leq 0 \\ \frac{1}{2}(1-\theta)^2 & 0 < \theta < 1 \\ 0 & \theta \geq 1 \end{cases}$$

where the loss function is smoothed when $\theta = 1$, and the corresponding optimization task of computing its gradient is more tractable. The gradient of the smoothed hinge loss is

$$h'(\theta) = \begin{cases} -1 & \theta \leq 0 \\ \theta - 1 & 0 < \theta < 1 \\ 0 & \theta \geq 1 \end{cases} \tag{5.5}$$

Next, we will introduce how to optimize the objective function in Eq.(5.4) efficiently.

5.3.4 Optimization

The objective function in Eq.(5.4) is not convex *w.r.t.* all three variables, *i.e.,* $\mathbf{U}, \mathbf{V}$, and $\mathbf{w}$. However, Eq.(5.4) is convex in each of the three variables separately.

Hence, we update each of them by fixing the other two iteratively.

Modeling Rumor Category

First, we introduce how $\mathbf{U}$ can be updated by fixing $\mathbf{V}$ and $\mathbf{w}$. By removing terms that are irrelevant to $\mathbf{U}$, Eq.(5.4) can be reformulated as follows:

$$\begin{aligned} &\min_{\mathbf{U}} \frac{1}{2}||\mathbf{D} - \mathbf{U}\mathbf{V}^T||_F^2 + \beta \sum_{i=1}^{m} h(\mathbf{u}_i \mathbf{V}^T \mathbf{w} \mathbf{y}_i). \\ &s.t. \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}, \mathbf{U} \geq \mathbf{0} \end{aligned} \tag{5.6}$$

The problem in Eq.(5.6) is an orthogonality constrained optimization problem. The problem can be solved in the Crank-Nicolson scheme. Following (Wen and Yin, 2013), $\mathbf{U}$ can be efficiently updated as follows:

$$\mathbf{U} \leftarrow (\mathbf{I} + \frac{\tau}{2}\mathbf{Q})^{-1}(\mathbf{I} - \frac{\tau}{2}\mathbf{Q})\mathbf{U}, \tag{5.7}$$

where τ is the step size and $\mathbf{Q}$ is a skew-symmetric matrix, which leads to the descent along geodesics and inside the feasible set. $\mathbf{Q}$ can be constructed as

$$\begin{aligned} \mathbf{Q} &= [\mathbf{U}, \mathbf{G}][\mathbf{G}, -\mathbf{U}]^T \\ &= \mathbf{U}\mathbf{G}^T - \mathbf{G}\mathbf{U}^T, \end{aligned} \tag{5.8}$$

where $\mathbf{G}$ is the gradient of the optimization objective in Eq.(5.6). Since both terms in Eq.(5.6) are convex, the gradient can be obtained with Eq.(5.5) as

$$\mathbf{G}_{i,j} = [\mathbf{U}\mathbf{V}^T\mathbf{V} - \mathbf{D}\mathbf{V}]_{i,j} + \beta[h'(\mathbf{u}_i \mathbf{V}^T \mathbf{w} \mathbf{y}_i)\mathbf{y}_i \mathbf{w}^T \mathbf{V}]_j,$$

where $[\cdot]_{i,j}$ is the (i, j) entry of the matrix and $[\cdot]_j$ is the j^{th} entry of the vector. A problem of directly updating Eq.(5.7) is that the time complexity is high, since the inverse operation dominates the calculation when m is large. In order to solve the problem, we rewrite the objective function as follows by applying the SMW formula (Sherman and Morrison, 1950; Wen and Yin, 2013):

$$\begin{aligned}
&(\mathbf{I}+\frac{\tau}{2}\mathbf{Q})^{-1}(\mathbf{I}-\frac{\tau}{2}\mathbf{Q})\mathbf{U} \\
&=(\mathbf{I}+\frac{\tau}{2}[\mathbf{U},\mathbf{G}][\mathbf{G},-\mathbf{U}]^T)^{-1}(\mathbf{I}-\frac{\tau}{2}[\mathbf{U},\mathbf{G}][\mathbf{G},-\mathbf{U}]^T)\mathbf{U} \\
&=\mathbf{U}-\tau[\mathbf{U},\mathbf{G}](\mathbf{I}+\frac{\tau}{2}[\mathbf{G},-\mathbf{U}]^T[\mathbf{U},\mathbf{G}])^{-1}[\mathbf{G},-\mathbf{U}]^T\mathbf{U}
\end{aligned} \tag{5.9}$$

By reformulating the objective function in Eq.(5.7), only the inverse of $(\mathbf{I}+\frac{\tau}{2}[\mathbf{G},-\mathbf{U}]^T[\mathbf{U},\mathbf{G}])$ needs to be calculated, which takes $O(k^3)$. Since k is the number of clusters and normally $k \ll n$ and $k \ll m$, the inverse operation is much easier to solve and no longer dominates the computation.

In order to find the optimal step size τ in Eq.(5.7), we first introduce the Armijo-Wolfe condition (Fletcher, 2013)

$$\mathcal{L}(\mathbf{U}_\tau) \leq \mathcal{L}(\mathbf{U}_{\tau=0}) + \rho_1 \tau \mathcal{L}'(\mathbf{U}_\tau), \tag{5.10}$$

$$\mathcal{L}'(\mathbf{U}_\tau) \geq \rho_2 \mathcal{L}'(\mathbf{U}_{\tau=0}),$$

where $\mathbf{U}_\tau$ is the trial point of gradient descent given a specific τ, and $\mathbf{U}_{\tau=0}$ is the value by setting τ to zero. ρ_1 and ρ_2 are two parameters satisfying that $0 < \rho_1 < \rho_2 < 1$ (Moré and Thuente, 1994). $\mathcal{L}(\cdot)$ is the loss function in Eq.(5.6), and $\mathcal{L}'(\cdot)$ is its gradient.

The optimal value of τ can be obtained through curvilinear search (Box *et al.*, 1969) with Armijo-Wolfe condition in Eq.(5.10), and details are presented in Algorithm 3.

Algorithm 3 Curvilinear Search for τ

```
1: Initialize $\tau > 0$
2: **Until** Eq.(5.10) is satisfied
3:     Set $\tau \leftarrow \frac{\tau}{2}$
4: **Return** $\tau$
```

Selecting Features

Now we are introducing how $\mathbf{V}$ can be updated given fixed $\mathbf{U}$ and $\mathbf{w}$. The optimization function of $\mathbf{V}$ can be formulated based on Eq.(5.4) as

$$\min_{\mathbf{V}} \frac{1}{2}||\mathbf{D} - \mathbf{U}\mathbf{V}^T||_F^2 + \alpha||\mathbf{V}||_{2,1} + \beta \sum_{i=1}^{m} h(\mathbf{u}_i\mathbf{V}^T\mathbf{w}\mathbf{y}_i), \tag{5.11}$$

where constraints on $\mathbf{U}$ are removed. The objective function in Eq.(5.11) is similar to that of multi-task feature selection (Obozinski *et al.*, 2006). The update rule for $\mathbf{V}$ can be obtained by taking the derivative and setting it to zero. The derivative can be formulated as

$$\mathbf{V} - \mathbf{D}^T\mathbf{U} + \alpha\mathbf{C}\mathbf{V} + \beta \sum_{i=1}^{m} (\mathbf{y}_i h'(\mathbf{u}_i\mathbf{V}^T\mathbf{w}\mathbf{y}_i))\mathbf{w}\mathbf{u}_i, \tag{5.12}$$

where $\mathbf{C}$ is a diagonal matrix where $\mathbf{C}_{i,i} = \frac{1}{2||\mathbf{v}_i||_2}$. $\mathbf{C}$ is constructed to obtain the derivative of the $\ell_{2,1}$ regularization term of $\mathbf{V}$ (Tang and Liu, 2012). By setting Eq.(5.12) to zero, the update rule of $\mathbf{V}$ can be written as:

$$\mathbf{V} \leftarrow (\mathbf{I} + \alpha\mathbf{C})^{-1}(\mathbf{D}^T\mathbf{U} - \beta \sum_{i=1}^{m} (\mathbf{y}_i h'(\mathbf{u}_i\mathbf{V}^T\mathbf{w}\mathbf{y}_i))\mathbf{w}\mathbf{u}_i). \tag{5.13}$$

Learning Rumor Classifier

Finally, we will introduce how the rumor classifier can be obtained given fixed $\mathbf{U}$ and $\mathbf{V}$. By removing terms that are irrelevant to $\mathbf{w}$, Eq.(5.4) can be rewritten as

$$\min_{\mathbf{w}} \beta \sum_{i=1}^{m} h(\mathbf{u}_i\mathbf{V}^T\mathbf{w}\mathbf{y}_i) + \frac{\gamma}{2}||\mathbf{w}||_2^2,$$

where we add a regularization term to avoid over-fitting, and γ to control the complexity of $\mathbf{w}$. Since both terms are smooth and convex, the update rule of $\mathbf{w}$ can be written as

$$\mathbf{w} \leftarrow \mathbf{w} - \eta\big(\beta(\mathbf{y}_i h'(\mathbf{u}_i \mathbf{V}^T \mathbf{w} \mathbf{y}_i))\mathbf{V}\mathbf{u}_i^T\big), \tag{5.14}$$

where η is the step size and can be efficiently estimated with backtracking line search (Nocedal and Wright, 2006).

Analysis

Given update rules of $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{w}$, the problem can be efficiently solved by a Stochastic Gradient Descent algorithm (SGD). SGD solves the optimization problem in the hill-climbing scheme by seeking the stationary point. The optimization process can be found in Algorithm 4. $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{w}$ are updated alternatively from line 3 to line 5. Since the objective function decreases for each of the subproblems, and Eq.(5.4) has lower bounds such as zero, Algorithm 4 converges. As mentioned earlier, the inverse operation in Eq.(5.9) can be quickly done in $O(k^3)$. The inverse operation in Eq.(5.13) can be solved in $O(n)$ since $(\mathbf{I} + \alpha\mathbf{C})$ is a diagonal matrix. Therefore, the complexity of one iteration (lines 3-5) is dominated by the matrix multiplication, which can be efficiently solved since the data matrix $\mathbf{D}$ obtained from social media contents is usually sparse. In addition, the experimental results on our datasets show that the algorithm often converges in less than 20 iterations.

5.4 Experiments

In this section, we conduct experiments to assess the performance of the proposed framework, namely Cross-topic Emerging Rumor deTection (CERT), with real world social media data. In particular, we aim to answer the following two questions through experiments:

Algorithm 4 Early Detection of Emerging Rumors

```
Input: Data matrix $\mathbf{D}$, label vector $\mathbf{y}$, maximal number of iterations $I$
Output: $\mathbf{U}$, $\mathbf{V}$, $\mathbf{w}$
Generate $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{w}$ randomly
For $i$=1 to $I$ do
    Update $\mathbf{U}$ by Eq.(5.9)
    Update $\mathbf{V}$ by Eq.(5.13)
    Update $\mathbf{w}$ by Eq.(7.8)
    If convergence Break
End For
Return $\mathbf{U}$, $\mathbf{V}$, $\mathbf{w}$
```

- How effective is CERT in detecting emerging rumors in social media by leveraging prior labeled data of rumors?
- How quickly can CERT detect emerging rumors after rumors start being spread with only prior labeled data of rumors?

We begin by introducing how we obtain the real-world social media data and the corresponding ground truth. Then we introduce the experimental setup and baselines for comparison. Based on the experimental results, we finally investigate the effectiveness and the earliness of CERT on rumor detection.

5.4.1 Datasets

Over 200 million posts are posted per day on Twitter[2] and the popularity has made Twitter a testbed for rumor detection research (Qazvinian *et al.*, 2011; Sampson *et al.*, 2016a; Zhao *et al.*, 2015). In this section, we aim to collect a large dataset that includes

[2] https://blog.twitter.com/2011/200-million-tweets-per-day

tweets about all prior rumors within a certain period. Following (Qazvinian *et al.*, 2011), we leverage Twitter Search API[3] to retrieve tweets of interests by compiling queries with a fact-checking website.

In order to validate and debunk unverified information, several fact-checking websites have been developed. Verification of rumors on fact-checking websites is mainly run by professional editors and trusted information sources. Though fact-checking sites may cover only a small portion of rumors in social media, the identified rumors offers us valuable resources to evaluate rumor detection algorithms. In this section, we choose Snopes[4] to obtain ground truth, which is the top rumor reference site according to Alexa[5]. In order to obtain non-rumor posts pertaining to the same topic, we extract keywords in regular expressions as queries to retrieve posts.

With queries generated from 252 rumors from June 30^{th} to July 11^{th}, we collect 9,918 tweets and hire two human annotators to manually verify that they are rumors. The annotators classify a tweet by reading the content and referring to the Snopes article. The inter-judge agreement over all data instances achieves a high Cohen's κ score 0.93, which demonstrates the annotation accuracy. An expert makes the final judge when annotators disagree with each other. The resultant dataset contains 1,618 rumor instances and 8,300 non-rumor instances.

5.4.2 Experimental Settings

We follow conventional settings (Qazvinian *et al.*, 2011) to evaluate the performance with *Precision*, *Recall*, and *F-measure*. All other parameters are set with cross-validation based on a holdout dataset. Next, we will introduce methods that

[3] https://dev.twitter.com/rest/public/search

[4] http://www.snopes.com/

[5] http://www.alexa.com/topsites/category/Society/Folklore/
Literature/Urban_Legends

we use to compare with CERT. First, we aim to investigate how effective is CERT in detecting emerging rumors with the historical rumors. Since CERT jointly clusters rumors, selects features and trains a classifier, first, we introduce three variants of the proposed method to validate different aspects of CERT:

- **Pooling** trains the classifier on the prior training data directly without clustering data or selecting features, and we adopt the linear-SVM as the classifier. As shown in Figure 5.1, the way of CERT to model prior labeled data is to cluster them into different rumor categories. On the contrary, Pooling directly learns a classifier with all prior labeled data. Hence, Pooling is used to validate the necessity of structure learning and feature selection.

- **Elastic Net** trains the classifier by imposing a sparsity regularization term to select features. Pooling aims to evaluate structure learning and feature selection as a whole, while Elastic Net only tests the effectiveness of feature selection. Elastic Net aims to learn a sparse classifier with fewer selected features without clustering data instances into rumor categories. So the result can be used to validate the necessity of structure learning.

- **KM_SVM** first clusters data instances and trains a classifier for each cluster. KM_SVM is designed to evaluate the method that separately clusters data and trains classifiers. Since we propose to unify the data clustering and classifier learning processes, the result of KM_SVM can be used to validate the necessity of the joint learning framework. Given a test instance, we first find the closest cluster center and apply the corresponding classifier of the cluster to determine the label of the test instance.

Several methods have been proposed to identify unverified information from social media. In order to compare with the state-of-the-art approaches, we include the following methods:

- **FE_LL** (Qazvinian *et al.*, 2011): Rumors that are widespread in social media usually share similar patterns in terms of content and diffusion. In order to capture the patterns of rumors, Qazvinian *et al.* implement a method to extract relevant features that capture the patterns of rumors. Based on the extracted features and labeled instances, classifiers are trained to predict rumors. The adopted classifier is a ℓ_1-regularized log-linear model.

- **LK_RBF** (Sampson *et al.*, 2016a): A problem that hinders the early detection of rumors is the data scarcity: only few comments are available and they are scattered in different discussion threads. In order to relieve the scarcity, a possible way is to combine these individual tweets from different threads together as a "conversation". Sampson *et al.* propose several methods to combine tweets and try different supervised learning methods to classify rumors. We choose the URL-based method to combine tweets and the RBF kernel method as the classifier, which achieve the best performance in that work and also on our dataset. LK_RBF is effective for detecting rumors in the early stage, and the comparison can be used to evaluate the earliness of CERT.

We design two experiments to show the performance of CERT. In the first experiment for studying the effectiveness, we arrange rumors in the chronological order by the starting time, and we take the first 50% for training and the rest for testing. Therefore, all methods predict new rumors with historical training data and the experiment shows the performance on cross-training. In the second one, baseline methods are trained on the rumors for evaluation, and the training data is added in the chronological order by the generation time. The second experiment shows the minimum time that could be saved by CERT regardless of the annotation.

Table 5.2: Performance on Detecting Emerging Rumors.

Approaches	Precision	Recall	F-score
Pooling	76.13%	60.20%	67.23%
Elastic Net	79.56%	65.62%	71.92%
KM_SVM	70.12%	72.55%	71.31%
FE_LL	86.29%	85.33%	85.81%
LK_RBF	80.16%	64.62%	71.56%
CERT	**92.18%**	**88.15%**	**90.12%**

5.4.3 Effectiveness Analysis

The comparison of the performance is shown in Table 5.2. Precision shows how accurate rumors can be detected, recall shows how sensitive the models are to rumors, and F-score (F-1 measure) is the harmonic mean of precision and recall. Based on the results shown in Table 5.2, we draw the following observations. The three variants, *i.e.,* Pooling, Elastic Net, and KM_SVM, cannot effectively detect emerging rumors with historical training data. Imposing a feature selection is useful since Elastic Net outperforms Pooling. Disjointly clustering and detecting rumors with KM_SVM does not achieve comparable results, which proves the necessity of a coherent method.

Among the two rumor detection methods, *i.e.,* FE_LL and LK_RBF, FE_LL achieves the better results and is the runner-up among all methods, showing that feature engineering helps detect rumors better. The feature engineering process can be integrated into CERT easily. CERT outperforms existing methods by jointly grouping data instances, selecting features and learning classifiers. The result empirically demonstrates that CERT is effective in exploiting knowledge in historical training data.

5.4.4 Earliness Analysis

In the second experiment, we allow existing rumor detection methods to be trained on rumors that are for evaluation. Through incrementally adding training data in the chronological order, we will be able to estimate the time that can be saved by utilizing historical data. The results on earliness are shown in Figure 5.2. Note that, CERT is trained **only** with historical data, meaning that when the other two methods are trained on more labeled data of the emerging rumor, CERT is not retrained and only exploits the prior labeled data.

At an early stage with 10% to 50% training data, LK_RBF outperforms FE_LL regarding F-score, showing that linking and combining posts with the same URLs alleviates the data scarcity problem. With more data being generated, the advantages of linking data become diminishing, and FE_LL outperforms LK_RBF. The result shows that FE_LL is more effective with abundant training data, while LK_RBF is more useful for an emerging rumor. However, the best baseline achieves the result of CERT with 70% training data, which has an average time lag of 22 hours. Therefore, we empirically prove that the use of CERT not only yields effective classifiers but also finds emerging rumors faster than existing approaches.

5.4.5 Rumor Categories

An intermediate task is to cluster rumors into categories, which is helpful for the detection since rumors of the same category trigger similar reactions (DiFonzo and Bordia, 2007). To help understand the clustering results, we show three example categories and the corresponding top rumors in the category. The results are illustrated in Table 5.3, including wedge-driving rumors, dread rumors and curiosity rumors. The name of the three clusters is acquired through manual checking. We see that rumors

Table 5.3: Three Example Categories of Rumors Detected by CERT.

Wedge-Driving Rumors	Dread Rumors	Curiosity Rumors
President Obama claimed that Americans would be better off under the martial law during an interview with Washington Post.	Police in assessed that an encounter with three men at Silver Lake Park was an attempted human trafficking incident.	A North Carolina provider of mental health services is named “Nutz R Us.”
A Black Lives Matter protest in Memphis obstructed I-40, leading to the death of a critically ill child transplant patient.	A “purge” event is planned for 9 July 2016 in Baton Rouge kill all police officers.	A fisherman captured a 3,000 lb. great white shark out of the waters in the Great Lakes Michigan.
A police officer shot two-year-old Malik Gibson after mistaking his pacifier for a gun.	NASA has warned of imminent disaster due to the trajectory of Nibiru.	Researchers sequenced octopus genomes and discovered alien DNA.

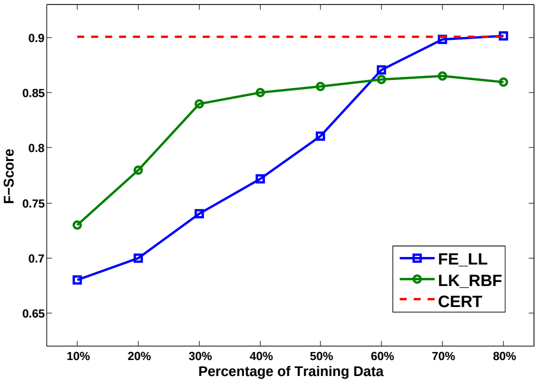


Figure 5.2: Performance of Traditional Approaches with Chronologically Additional Training Data, While CERT Uses the Historical Data.

are clustered cohesively, and the cohesiveness explains how it facilitates selecting key features from sparse data.

5.5 Summary

Circulating online rumors have become a key issue for today’s social media sites. They may result in catastrophic effect both online and offline quickly. After they go viral, it is extremely difficult to eliminate their existence. In order to detect rumors at an early stage, we propose to directly train a classifier based on readily available labeled data from prior rumors. Motivated by traditional studies on rumors, we introduce a novel framework that jointly clusters data, selects features, and trains classifiers. An optimization approach is also presented to solve the problem efficiently. The proposed framework, CERT, largely breaks the bottleneck of the time lag from annotating datasets. Experimental results illustrate the effectiveness and earliness of CERT on real-world data.

Chapter 6

CLASSIFICATION WITH MISINFORMATION

In this chapter, I study the problem of mitigating the negative effect of misinformation on a machine learning algorithm. I focus on statistical relational learning, which is mostly based on social media data and has been widely applied. I will first introduce the emerging challenges brought by misinformation. In addition, I will present the proposed method and evaluations based on real-world data.

6.1 Challenges of Misinformation Contaminating Social Media Data

Relational learning (RL) utilizes relationships between instances manifested in a network to improve the predictive performance of various network mining tasks. The triumphant applications of RL have been witnessed in a myriad of domains, such as social networks (*e.g.*, Flicker), language networks (*e.g.*, Wikipedia), and citation networks (*e.g.*, DBLP). The vast amount of social media content, ranging from daily chatter, conversations to information sharing and news reports, together with automatic modeling of the content information, allow for an insight into the public opinion that has been utilized for recommender systems (Guy *et al.*, 2010), targeted advertising (Tucker, 2014), and even predicting the stock price (Bollen *et al.*, 2011) and election results (Tumasjan *et al.*, 2010).

However, due to emerging challenges brought by malicious social media users, it is increasingly risky to depend on social media data for decision making. Most social media platforms are open to register and easily accessible, which enables malicious users to spread misinformation while easily disguise their accounts. For example, thousands of bot accounts were found to intentionally spread misinformation during the 2016 U.S. election[1]. To complicate the problem, in order to avoid being detected, they copy legitimate content from normal users (Wu *et al.*, 2017a) and farm links with other people (Hooi *et al.*, 2016). The manipulated content and links camouflage the malicious users that further lead to a polluted dataset on which decision makers may rely to design public policies.

In this section, we precisely focus on the computational challenge brought by emerging misinformation in social media data. Existing efforts in this area mainly focus on a deletion-based way to solve the problem: building a detection model to identify polluted points, removing them from the data, and learning a predictive model with the refined dataset. However, the ground truth data for the malicious users itself can be very difficult to obtain. Hence, a deletion-based method is limited by the availability of additional label information. In real applications, for sake of simplicity, noisy data are often directly used. Therefore, it would be appealing if the negative effect of noisy data instances can be seamlessly mitigated.

The task of learning a predictive model in the presence of misinformation is particularly difficult, if not impossible, especially when we are lacking availability of labels of malicious users. In order to tackle the challenge, we assume that the real performance can be tested on a holdout dataset, and the optimal performance can be achieved by selecting only the unpolluted data instances. Therefore, an optimal set of model coefficients can be achieved by exhausting all possible combinations of instances. Given the size of the selected instance set, the task is a NP-hard problem due to the combinatorial property. Since the size is also a variable and the size of a dataset is usually very large, it is computationally unfeasible to directly search for it. To this end, we propose a novel relational learning method, Relational Learning with Misinformation (RLM), to identify the set of instances in polynomial time.

[1] https://www.nytimes.com/2017/09/07/us/politics/russia-facebook-twitter-election.html

In order to tackle the computational challenge, we utilize the social network structure to facilitate the search for optimal coefficients. As revealed in social identity theory (Hogg, 2016), the membership of social community is likely to indicate the similar identity shared among all community members, and the community structure is relatively less susceptible to be affected by malicious behaviors. Hence, we propose to model the community structure with an adaptive group Lasso approach to solve the instance selection problem for relational learning.

6.2 Problem Statement

Given a set of social media users, and consider $\mathbf{V} \in \mathbb{R}^{m \times n}$ is the attribute matrix where m is the number of users and n is the number of features, $\mathbf{P} \in \mathbb{R}^{m \times m}$ denotes the adjacency matrix manifested by the social network structure where $\mathbf{P}_{i,j} = 1$ indicates that user i follows j and it equals to 0 otherwise, $\mathbf{t} \in \{0, 1\}^m$ is a label vector represents whether a user contains a certain social tag. Given label information for a subset of users $\mathbf{t} \in \{0, 1\}^{m_{tr}}$, due to the influence of misinformation, the label vector is noisy and thus there are k instances mislabeled, we aim to predict labels for the rest m_{te} unlabeled users where $m = m_{tr} + m_{te}$. More formally, the problem is stated as follows:

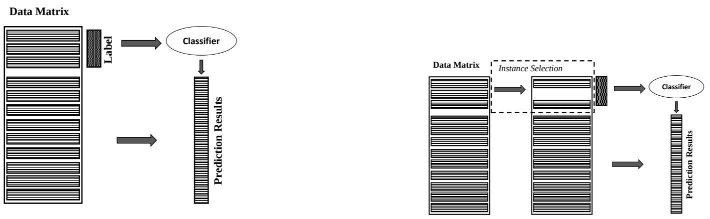
Input

a user-attribute matrix $\boldsymbol{V}$, an adjacency matrix $\boldsymbol{P}$ and the label information $\boldsymbol{y}^{tr} = \{0, 1\}^{m_{tr}}$ for a subset of m_{tr} users.

Output

labels of test users, $\boldsymbol{t}^{m_{te}} = \{0, 1\}^{m_{te}}$, where m_{te} is the size of testing data.

A mislabeled instance indicates that the label fails to reveal the true identity of the user. In the process of learning, we posit the existence of misinformation and



(a) Relational learning with social media users.

(b) Relational learning with selected instances.

Figure 6.1: Illustration of Comparison Between Traditional Relational Learning and the Proposed Approach with Instance Selection. A Classic Relational Learning Method Directly Constructs a Classifier with Available Label Information; While the Proposed Framework First Removes Noise from the Label Information by Actively Selecting Instances, upon Which a Classifier Is Built.

aim to select top k instances that are not mislabeled to build an optimal predictive model. The social labels can be obtained from different sources on different platforms. For example, Flickr users can join different groups and BlogCatalog users are able to subscribe and add tags for themselves. The group memberships and interest tags can be extracted as labels.

6.3 Robust Statistical Relational Learning with Misinformation

In order to illustrate our intuition, we illustrate the framework of classic relational learning and the proposed approach in Figure 6.1. A conventional practice of dealing with social media network data is to construct a classifier with the data matrix extracted from users. Considering potential negative effect brought by misinformation, we argue a model with better accuracy can be obtained by selecting a subset of instances for training. As shown in Figure 6.1, an additional instance selection module is introduced.

Social networking platforms allow users to freely post content information, which reveals the preference and interests of a user and thus could be utilized to characterize the user in relational learning. To this end, a classifier can be constructed by minimizing:

$$\frac{1}{2}||\mathbf{V}\mathbf{w}-\mathbf{y}||_2^2, \tag{6.1}$$

where $\mathbf{V} \in \mathbb{R}^{m \times n}$ is the data matrix, and m is the number of users and n is the number of textual features. Linear regression is adopted here for generality, and $\mathbf{w} \in \mathbb{R}^n$ represents the model coefficients that need to be optimized. $\mathbf{y} \in \mathbb{R}^m$ is the label vector of training data. Throughout the paper, we focus on a binomial classification setting which can be easily extended to the multinomial case.

In order to avoid over-fitting, a regularization term is often adopted to control the model complexity. The model can then be formulated as:

$$\frac{1}{2}\min_{\mathbf{w}} ||\mathbf{V}\mathbf{w}-\mathbf{y}||_2^2 + \frac{\lambda_1}{2}||\mathbf{w}||_2^2, \tag{6.2}$$

where λ_1 controls the cutoff between model complexity and accuracy. A larger λ_1 leads to a more simplified model. The formulation achieves an optimal $\mathbf{w}$ through minimizing the training error. Considering the negative effect of misinformation, we introduce to integrate instance selection as,

$$\begin{aligned} &\min_{\mathbf{w},\mathbf{c}} \frac{1}{2}\sum_{i=1}^{m} \mathbf{c}_i(\mathbf{V}_{i,*}\mathbf{w}-\mathbf{y}_i)^2 + \frac{\lambda_1}{2}||\mathbf{w}||_2^2 \\ &\text{subject to} \quad \sum_i \mathbf{c}_i = k, \mathbf{c} \in \{0,1\}^m, \end{aligned} \tag{6.3}$$

where we introduce an instance selection term $\mathbf{c} \in \{0,1\}^m$ to select k instances to only have influence on the classifier, and k is a predefined budget. Due to the combinatorial nature, it is an *NP-hard* problem which can be difficult to solve. It could also be laborious to find an optimal k. In order to cope with the computational challenge, we try to leverage the social network structures.

On a social networking site, users can be organized by assorted social groups and communities. Since the community structure is often induced from the homophily or proximity relationship between users, it provides a valuable perspective of user profiles (Hogg, 2016). Here, we posit the correlation between social community structure and the information quality that, users belonging to the same group are more likely to provide content of similar quality. The community structure is also more robust to the link farming of malicious users: randomly establishing a link with a legitimate user can be relatively easy, while establishing links with multiple users belonging to the same community can be very difficult.

Next, we define an *index tree* to denote the social community structure for brevity of presentation,

Definition 1 ***Index tree****: Let T denote a tree of depth d, where non-leaf nodes represent social communities and leaf nodes are users. Let $T_i = \{G^i_1, G^i_2, \ldots, G^i_{n_i}\}$ denote the nodes on layer i, where $n_0 = 1$ and n_i is the number of nodes on layer i. Given $i < d$, G^i_j represents j^{th} group on the i^{th} layer. $G^0_1 = \{1, 2, \ldots, m\}$ contains indices of all users. In order to maintain a tree structure, nodes should satisfy the following conditions: 1) Nodes on the same layer share no indices with each other ($G^i_j \cap G^i_k = \emptyset, \forall i = 0, \ldots, d, j \neq k, j \leq n_i, k \leq n_i$); 2) Given a non-root node G^i_j, we denote its parent node as G^{i-1}_{j0} ($G^i_j \subseteq G^{i-1}_{j0}, 1 < i \leq d$).*

In order to obtain such a group structure, we select a hierarchical community detection method, namely Louvain (Blondel *et al.*, 2008), where maximum modularity is used to optimize the group structure. The code is available[2].

Given a social community structure, the task of instance selection can boil down to community selection. Though the search space is significantly reduced, exhausting all possible combinations can also be time-consuming. To this end, we further relax

[2] https://perso.uclouvain.be/vincent.blondel/research/louvain.html

the constraint on $\mathbf{c}$ and rewrite the optimization objective in Eq.(6.3),

$$\begin{aligned} &\min_{\mathbf{w},\mathbf{c}} \frac{1}{2}\sum_{i=1}^{m} \mathbf{c}_i(\mathbf{V}_{i,*}\mathbf{w} - \mathbf{y}_i)^2 + \frac{\lambda_1}{2}||\mathbf{w}||_2^2 + \lambda_2 \sum_{i=0}^{d}\sum_{j=1}^{n_i} ||\mathbf{c}_{G_j^i}||_2 \\ &\text{subject to} \quad \sum_i \mathbf{c}_i = k, \end{aligned} \tag{6.4}$$

where we relax the instance selection vector $\mathbf{c}$ to be a non-binary vector. In order to make the vector more "binary" to align it with the objective of instance selection, we propose to force more entries in $\mathbf{c}$ to be exact 0 or 1. Specifically, we integrate a structured sparsity regularizer $||\mathbf{c}_{G_j^i}||_2$. λ_2 is used to control the extent of sparsity. The adopted sparsity regularizer is a tree-structured group Lasso (Hastie *et al.*, 2015),

$$\sum_{i=0}^{d}\sum_{j=1}^{n_i} ||\mathbf{c}_{G_j^i}||_2, \tag{6.5}$$

where an ℓ_2-*norm* is imposed on each member of a group, and an ℓ_1-*norm* is imposed on weights of all groups. This ℓ_{21}-*norm* is iteratively imposed on the social community structure in a bottom-up manner. The combination of ℓ_1- and ℓ_2-*norm* leads to sparse representation of $\mathbf{c}$, while ℓ_1-*norm* determines the organization of sparsity (Meier *et al.*, 2008). In particular, imposing ℓ_1-*norm* within each group leads to the inter-group sparsity, *i.e.,* weights of users in some groups are selected to be assigned higher weights, while users in other groups are with lower weights. Therefore, by minimizing the training error, groups that lead to better accuracy are selected by the sparse representation of $\mathbf{c}$.

6.3.1 Optimization

In this section, we introduce how we optimize the problem efficiently. Two variables need to be optimized in Eq.(6.4), $\mathbf{c}$ for instance selection and $\mathbf{w}$ for classifying users. The problem is not jointly convex *w.r.t.* both variables simultaneously. As a conventional practice, we alternatively optimize one variable by fixing the other.

The optimization problem boils down to two convex optimization tasks, and we keep iterating over them until convergence.

Instance Selection

Here we focus on optimizing $\mathbf{c}$ while keep $\mathbf{w}$ being fixed. Since the squared loss $(\mathbf{V}\mathbf{w} - \mathbf{y}_i)^2$ becomes a constant, we replace it with $\mathbf{p}$, where $\mathbf{p}_i = (\mathbf{V}_{i,*}\mathbf{w} - \mathbf{y}_i)^2$. The objective can then be reformulated as:

$$\begin{aligned} &\min_{\mathbf{c}} \frac{1}{2}\sum_{i=1}^{m} \mathbf{c}_i \mathbf{p}_i + \lambda_2 \sum_{i=0}^{d}\sum_{j=1}^{n_i} ||\mathbf{c}_{G_j^i}||_2 \\ &\text{subject to} \quad \sum_i \mathbf{c}_i = k, \end{aligned} \tag{6.6}$$

where the regularizer $||\mathbf{w}||_2^2$ that is fixed here is also omitted. It is easy to prove that Eq.(6.6) is strongly convex but not directly differentiable, *i.e.*, it is convex and non-smooth with respect to $\mathbf{c}$. In order to find the solution for the optimization problem in Eq.(6.6), we reformulate the problem as follows:

$$\phi_{\lambda_2}(\mathbf{c}) = \arg\min_{\mathbf{c}} \frac{1}{2}||\mathbf{c} - \mathbf{x}||^2 + \lambda_2 \sum_{i=0}^{d}\sum_{j=1}^{n_i} ||\mathbf{c}_{G_j^i}||_2, \tag{6.7}$$

where $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{x}_i = \frac{\mathbf{p}_i^{-1}}{\sum_k^m \mathbf{p}_k^{-1}}$. Therefore, the equality constrained optimization problem is transformed to a Moreau-Yosida regularization problem with the euclidean projection of $\mathbf{c}$ on to a vector $\mathbf{x}$ (Lemaréchal and Sagastizábal, 1997). The new formulation is continuously differentiable and it admits an analytical solution (Liu and Ye, 2010). Given a proper λ_2, the optimal $\mathbf{c} \in \mathbb{R}^m$ can be obtained in an agglomerative manner, which is shown in Algorithm 6. In the algorithm, the superscript of $\mathbf{c}$ is used to denote the layer of the tree, meaning that the output of the algorithm is $\mathbf{c}^0$. The bisection method can be implemented to find the optimal λ_2. Empirically, λ_2 can be initialized as $\sqrt{\frac{||l'(\mathbf{0})||_2^2}{\sum_{i=0}^{d} n_i}}$, where $l(c) = \frac{1}{2}||\mathbf{c} - \mathbf{x}||^2$. Then we use $\phi_{\lambda_2}(-l'(\mathbf{0}))$ to test whether λ_2 achieves the certain threshold. When $\phi_{\lambda_2}(-l^{'}(\mathbf{0})) = \mathbf{0}$, which means λ_2 is large enough to generate a trivial solution, we start looking for the lower bound as follows:

$$\lambda_2^{(lower)} = \max\{\lambda_2^{(i)} | \lambda_2^{(i)} = \frac{\lambda_2^{(i)}}{2^i}, \pi_{\lambda_2^{(i)}}(-l^{'}(\mathbf{0})) \neq \mathbf{0}\} \tag{6.8}$$

otherwise, if $\phi_{\lambda_2}(-l^{'}(\mathbf{0})) \neq \mathbf{0}$, we start looking for the upper bound as follows:

$$\lambda_2^{(upper)} = \min\{\lambda_2^{(i)} | \lambda_2^{(i)} = 2^i \lambda_2^{(i)}, \pi_{\lambda_2^{(i)}}(-l^{'}(\mathbf{0})) = \mathbf{0}\} \tag{6.9}$$

Algorithm 5 Solution of Moreau-Yosida Regularization

```
Input: $\{\mathbf{c}, G, \lambda_2\}$
Output: $\mathbf{c}^0$.
1: Set $\mathbf{c}^{d+1} = \mathbf{x}$,
2: for $i = d$ to $0$ do:
3:     for $j = 1$ to $n_i$ do:
4:         Compute:
           $\mathbf{c}^i_{G^i_j} = \begin{cases} \mathbf{0} & if \quad ||\mathbf{c}^{i+1}_{G^i_j}||_2 \leq \lambda_2, \\ \frac{||\mathbf{c}^{i+1}_{G^i_j}||_2 - \lambda_2}{||\mathbf{c}^{i+1}_{G^i_j}||} \mathbf{c}^{i+1}_{G^i_j} & if \quad ||\mathbf{c}^{i+1}_{G^i_j}||_2 > \lambda_2, \end{cases}$
5:     end for
6:end for
```

In Algorithm 6, we traverse the tree in an agglomerative manner, *i.e.*, from leaf nodes to the root node. At each node, the ℓ_2-*norm* of the weight $\mathbf{c}$ can be reduced by at most λ_2 as shown in step 4. After the traverse, the analytical solution of $\mathbf{c}$ can be achieved.

Predictor Training

When $\mathbf{c}$ is fixed, the problem only depends on $\mathbf{w}$. We reformulate the objective function as follows:

$$\epsilon_{\mathbf{w}} = \frac{1}{2}\sum_{i=1}^{m} \mathbf{c}_i(\mathbf{V}_{i,*}\mathbf{w} - \mathbf{y}_i)^2 + \frac{\lambda_1}{2}||\mathbf{w}||_2^2. \tag{6.10}$$

Therefore, the problem is reduced to an ℓ_2 regularized weighted linear regression problem, which is to minimize the cost $\epsilon_{\mathbf{w}}$. Since social media users and their corresponding contents may be massive, a scalable optimization method is needed. Here we use Stochastic Gradient Descent (SGD) (Bottou, 2010). Since Eq.(6.10) is convex, the corresponding gradient can directly be obtained as:

$$\frac{\partial \epsilon_{\mathbf{w}}}{\partial \mathbf{w}} = \sum_{i=1}^{m} \mathbf{c}_i \mathbf{V}_{i,*}^T(\mathbf{V}_{i,*}\mathbf{w} - \mathbf{y}_i) + \lambda_1 \mathbf{w}. \tag{6.11}$$

SGD is scalable since data examples can be updated in parallel (Zinkevich *et al.*, 2010). Detailed discussions about the performance can be found in Section 3.5.

6.3.2 Time Complexity Analysis

Here we analyze the time complexity of the algorithm. The computational costs include computation of $\mathbf{c}$ and $\mathbf{w}$. The computational cost for $\mathbf{c}$ comes from estimating the Moreau-Yosida regularization problem, which takes $\sum_{i=0}^{d}\sum_{j=1}^{n_i}|G_j^i|$. The computation of $\mathbf{w}$ is a standard ℓ_2 regularized regression problem, which can be accelerated with the parallel implementation. The calculation of Louvain method could also speed up and it needs to be done only once as preprocessing (Blondel *et al.*, 2008). Since the optimization is conducted in an alternative manner and both sub-tasks are convex, both procedures will monotonically decrease. In addition, since the objective function has lower bounds, such as zero, the above iteration converges.

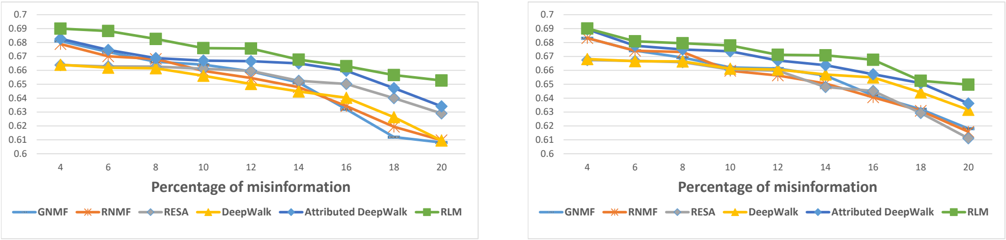


Figure 6.2: Comparison of Different Methods on the BlogCatalog Dataset with *Macro*-F_1 and *Micro*-F_1 Measures. Additional Training Instances are Randomly Selected and Flipped with the Label.

6.3.3 Convergence Analysis

Here we analyze the convergence condition. Since the optimization is conducted in an alternative manner and both sub-tasks are convex, both procedures will monotonically decrease. In addition, since the objective function has lower bounds, such as zero, the above iteration converges.

6.4 Experiments

RLM is proposed to seamlessly mitigate the negative effect of misinformation in a relational learning method. In this section, we aim to answer two research questions:

- How effective is the proposed method compared with other approaches in terms of classification accuracy?
- In the presence of misinformation, can the proposed RLM identify and downweight the anomalous training instances?

To answer the questions, we conduct experiments on two real-world social media datasets. Next, we will introduce the adopted datasets and experimental settings.

Table 6.1: The Statistics about Employed Datasets.

	# of Instances	# of Labels	# of features
BlogCatalog	5198	6	8189
Flickr	7575	9	12047

6.4.1 Datasets

We conduct experiments on two real-world social media datasets that are publicly available[3]. Table 6.1 illustrates some statistics about the two datasets. The users are randomly sampled from the two websites. Assorted features are extracted, such as text and scalar features like age. Following previous work (Perozzi *et al.*, 2014; Wu *et al.*, 2016a), we adopt the user interest tags in BlogCatalog and group memberships in Flickr as labels.

6.4.2 Baseline Methods and Metrics

Our work focuses on classifying instances in a graph. Therefore, we compare with state-of-the-art classification methods with content and network information. We follow experimental settings of graph representation learning approaches by learning a classifier upon the learned dimensions.

- *Graph Regularized NMF*: aims to utilize both content and network information to characterize attributed graph nodes (Cai *et al.*, 2011). Based on the assumption of homophily, connected nodes are regularized to be predicted with similar labels. We denote the method as *GNMF*.

- *Robust NMF*: In order to deal with the anomalous instances in a dataset, in the area of robust statistics. We adopt Correntropy Induced Metric Non-Negative Matrix Factorization (Du *et al.*, 2012) which extends NMF by incorporating

[3] http://socialcomputing.asu.edu/

a correntropy induced metric to mitigate the negative effect of non-Gaussian noise. The method is denoted as *RNMF*.

- *Relational Learning with Social Status*: Our previous work that particularly focuses on modeling social network users by integrating social status into the relational learning framework. We denote the approach as *RESA*.
- *DeepWalk*: is a state-of-the-art graph embedding algorithm that learns distributed representations of social network users, which reports optimal accuracy on the BlogCatalog and Flickr datasets (Perozzi *et al.*, 2014).
- *Attributed DeepWalk*: extends DeepWalk by jointly considering the attribute information of graph nodes and reports optimal results among a variety of methods on learning attributed graphs (Yang *et al.*, 2015).

6.4.3 Experimental Settings

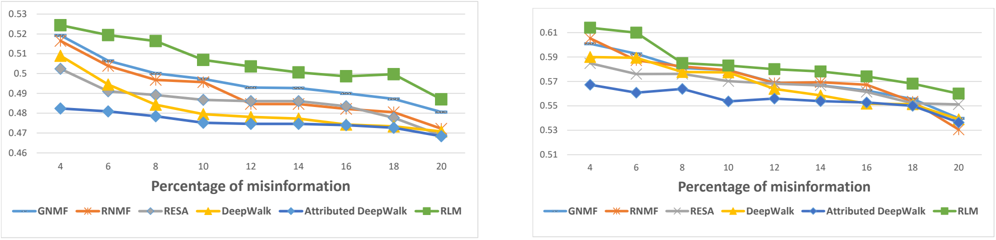
To test the prediction accuracy in terms of both precision and recall, we adopted the F_1-measure to evaluate the performance. Since the adopted dataset contains multiple class labels, and the instance number of different class labels is unbalanced, we adopt *Macro-*F_1 and *Micro-*F_1 to evaluate the performance of different methods.

Macro-F_1 is the arithmetic average of all classes, and it can be formulated as,

$$Macro - F_1 = \frac{1}{|\boldsymbol{T}|} \sum_{t \in \boldsymbol{T}} F_1^t, \tag{6.12}$$

where $\boldsymbol{T}$ is the set of all identity labels and F_1^t is the F_1-measure of task t.

A possible problem of Macro-F_1 is, since the size of different labels varies, the task with fewer instances may be overemphasized. Therefore, *Micro-*F_1 is adopted to



(a) The *Macro*-F_1 Measure of Different Methods on Flickr Data with Varying Percentage of Misinformation.

(b) The *Micro*-F_1 Measure of Different Methods on Flickr Data with Varying Percentage of Misinformation.

Figure 6.3: Comparison of Different Methods on the Flickr Dataset With *Macro*-F_1 and *Micro*-F_1 mEasures. Additional Training Instances Are Randomly Selected and Flipped with the Label.

mitigate the effect. First, we calculate the micro-averaged precision and recall:

$$Micro - precision = \frac{\#TP}{\#TP + \#FP} \tag{6.13}$$

$$Micro - recall = \frac{\#TP}{\#TP + \#FN}, \tag{6.14}$$

where #TP is the number of true positives, #FP is the number of false positives and #FN is the number of false negatives. Then Micro-F_1 is the harmonic average of Micro-precision and Micro-recall. In addition, five-fold cross-validation is adopted for all experiments, and the reported results are the average of all five folds.

In order to study the effect of misinformation, we randomly select instances in the training set to flip their labels. The classification is conducted in a One versus All (OvA) setting, so flipping the label means changing the label value to the opposite, *i.e.,* 0 to 1 or 1 to 0. Based on the modified training dataset, we learn the classifier and report the experimental results.

6.4.4 Experiments on BlogCatalog Data

The performance of different methods on BlogCatalog dataset with varying percentage of flipped instances, from 4% to 20%, is illustrated in Figure 6.2. The *x-axis* denotes the percentage of flipped instances, which are randomly sampled from the training set. From the experimental results we draw following observations:

- The proposed approach RLM outperforms all baselines in both settings. The margin between RLM and the runner-up models varies with different percentage of mislabeled data instances.

- The performance of Attributed DeepWalk is the runner-up method in both settings, which implies that both network and content information is useful in modeling a user.

- Since the class distribution of BlogCatalog data is relatively less skewed, the *Macro*- and *Micro*-F_1 results do not show drastic differences.

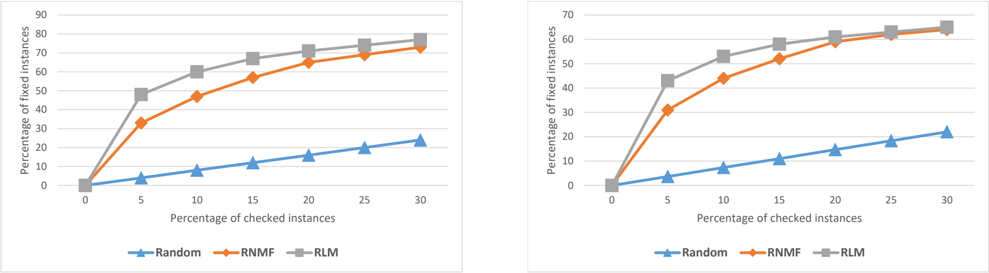
- DeepWalk has the lowest *Micro*- and *Macro*-F_1 among all six methods. Since DeepWalk investigates only the network information, the result reveals that content information is vital in characterizing social media users.

6.4.5 Experiments on Flickr Data

The performance of different methods on Flickr dataset is illustrated in Figure 6.3. Based on the experimental results, we draw following observations,

- The proposed RLM achieves the optimal *Macro*-F_1 (Figure 6.3(a)) and *Micro*-F_1 (Figure 6.3(b)) on the Flickr dataset.

- Different from the results of BlogCatalog, GNMF is the runner-up for *Macro*-F_1 and RNMF is the runner-up for *Micro*-F_1. Based on the definitions of *Macro*- and *Micro*-F_1, the result indicates that RNMF performs better at a class with more data instances, while GNMF performs relatively better on more *smaller* classes.



(a) The *Macro*-F_1 Measure of Different Methods on Flickr Data with Varying Percentage of Misinformation.

(b) The *Micro*-F_1 Measure of Different Methods on Flickr Data with Varying Percentage of Misinformation.

Figure 6.4: Comparison of Effectiveness of Different Methods in Identifying Mislabeled Instances for BlogCatalog and Flickr Datasets. Plots Show the Percentage of Mislabeled Nodes Being Fixed by Checking Instances in Training Data. RLM Ranks Data Instances with the Learned Weight in a Descending Order, RNMF Ranks Data with the Training Loss, and We Adopt a Random Baseline That Selects Nodes at Random.

- The runner-up method for BlogCatalog, Attributed DeepWalk, is with a relatively low F_1 measure on the dataset of Flickr. The method assumes nodes in the same latent community are more likely to have similar representations. However, since label information for Flickr is the group memberships, it is likely that users form a group without having similar interests or similar content, which contradicts the assumption of Attributed DeepWalk.

- The *Macro*-F_1 measure is generally better than the *Micro*-F_1 measure of all methods. Since we randomly select training instances without considering the class distribution, these minority classes are more vulnerable to the *flipping attacks*.

6.4.6 Analysis for Instance Selection

In this section, we study how well the proposed RLM can identify the mislabeled data instances. We use different methods to select suspicious data instances that are more likely to have been flipped. RLM downweights instances that are more likely to contain misinformation, so the weights are used to rank all data instances in a descending order. RNMF also directly models the negative effect of noisy data points, which is also adopted here. A baseline of Random is also introduced for comparison purposes, which selects instances at random. The results in Figure 6.4 show that adopting RLM allows us to efficiently find the mislabeled data points without checking too many instances, outperforming the other two baselines.

6.5 Summary

The massive amount of social media data allows automatic modeling of users in the social media network. Relational learning, which particularly focuses on interconnected data instances, have been successfully applied in a myriad of applications. An emerging challenge of utilizing social media data is the negative effect brought up by the misinformation. In this section, we precisely focus on the problem of mitigating its harm. In particular, we propose a unified framework that simultaneously selects data instances and learn a relational learning model. In order to allow for efficient optimization, we utilize the social community structure to effectively find groups of instances. We also transform the combinatorial problem into a convex optimization problem with relaxations. Experimental results on real-world datasets show the superiority of the proposed approach over competitive baseline methods. We also conduct experiments to understand how RLM selects and downweights data instances.

Chapter 7

PERSONALIZATION IN PRESENCE OF MISINFORMATION

In this chapter, I study the problem of personalizing social newsfeed with content and contextual information. I focus on optimizing both accuracy and earliness of the method, aiming at avoiding attention to be distracted by misinformation at an early stage. I will first review background of the problem. Second, I will formally define the task and present the proposed method. Experiments are presented based on real-world data over the state-of-the-art methods.

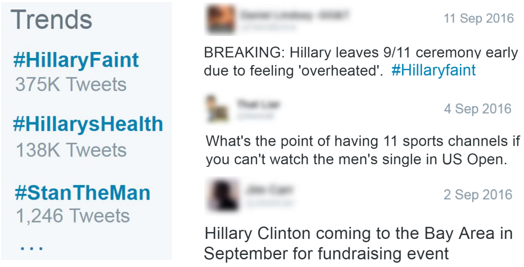
7.1 Emerging Challenges of Personalization in Social Media

Microblogging has become a main platform for dissemination of emerging issues, and some news broke out on Twitter[1] even before CNN. A recent study shows that 62% of American adults get news on social media[2]. Since various topics are trending simultaneously, it is critical to find a tailored list catering to users' interests. In this section, we aim to present a personalization system that tailors a personalized list of trending topics that are interesting to read for social media users.

A vital feature of trending topic personalization is its earliness. For example, the best timing to recommend topics for a baseball game is when it is ongoing since the stories become outdated soon after the match ends. Traditional approaches for personalization are incapable of dealing with trending topics since they rely on the accumulation of training data, such as contents for content-based filtering, and user-item interactions for collaborative filtering. For trending topics, both kinds of data are

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

[2] http://www.journalism.org/2016/05/26/news-use-across-social-media-platforms-2016/



Trends

#HillaryFaint
375K Tweets

#HillarysHealth
138K Tweets

#StanTheMan
1,246 Tweets

...

11 Sep 2016

BREAKING: Hillary leaves 9/11 ceremony early due to feeling 'overheated'. #Hillaryfaint

4 Sep 2016

What's the point of having 11 sports channels if you can't watch the men's single in US Open.

2 Sep 2016

Hillary Clinton coming to the Bay Area in September for fundraising event

Figure 7.1: Example User Posts and Trending Topics on September 11^{th}, 2016. The First Post Explicitly Includes the Trending Topic Hashtag, and the Later Two Are past Posts of Two Users.

generated with the topic going viral and becoming less attractive to read. Therefore, a key challenge of early personalization is to solve the cold-start problem.

Meanwhile, the auxiliary information is pervasively present on social networks. An auxiliary data source is the historical posts of users. Figure 7.1 shows an example of user posts and Twitter trending topics on September 11^{th}, 2016. There were over 600 trending topics on Twitter that day in the United States, including “HillaryFaint” and “HillarysHealth” that were about Hillary Clinton’s health issues[3], and “StanTheMan” which was about the US Open 2016 final[4]. The preferences of the first user can be easily found because of the post. But for the second and third user, the interests can be easily found only if their past posts can be used, since the second user posted on men’s single of US Open, and the third user was interested in Hillary’s upcoming fundraising trip. Another auxiliary data source is the links between users. “Birds of a feather flock together”, the principle of homophily reveals that friends on social networks are more likely to be interested in similar topics. A nice property of both kinds of auxiliary information is that they exist before a trending topic starts emerging, which can help solve the cold-start problem.

[3] http://www.foxnews.com/politics/2016/09/11/hillary-clinton-has-medical-episode-at-911-ceremony-source-says.html

[4] http://www.npr.org/2016/09/12/493563737/stan-wawrinka-beats-defending-u-s-open-champion-novak-djokovic

However, the auxiliary content information is hard to deal with. As shown in Figure 7.1, we lack labels for those posts that reveal user interests. Since the majority of user posts are irrelevant with a particular topic, adopting all posts would unavoidably introduce noise. Also, it would be time-consuming and costly to annotate these posts from a large number of posts manually. Therefore, the desirable method should be able to automatically identify and exploit the related posts for personalizing trending topics.

In this section, we present a graph-regularized multiple instance learning framework, Trending Topic Personalization approach (TTP), to personalize trending topics in an early stage. To solve the cold-start problem, TTP leverages social network structures to find the historical posts from like-minded users that would be useful for enriching the content of trending topics and preferences of users. To the best of our knowledge, this is the first work investigating personalization of trending topics on microblogging platforms.

7.2 Problem Statement

Let $\mathbf{U}$ denote the user set $\mathbf{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$. m represents the number of users and each user has a set of posts $\mathbf{u}_i = \{\mathbf{p}_{i1}, \ldots, \mathbf{p}_{i|\mathbf{u}_i|}\}$. Each post is an attribute vector, *i.e.*, $\mathbf{p}_{ij} \in \mathbb{R}^n$, where n is the number of textual features. $\mathbf{y} \in \{-1, 1\}^m$ is the label vector denoting whether a user is interested in a topic. Given a trending topic, $\mathbf{y}_i = 1$ (user i is interested in the topic) if one of i's posts contains the hashtag of the trending topic, and $\mathbf{y}_i = -1$ otherwise. Let $\mathcal{A}$ denote the set of social links between microblogging users, where $a_{ij} = 1$ if i follows j and $a_{ij} = 0$ otherwise. We now formally define the problem of personalizing trending topics as follows:

Given a trending topic, users $\mathbf{U}$*, the network information* $\mathcal{A}$*, and partial labels for training data* $\mathbf{y}$*, our goal is to learn an optimal function* f *that accurately predicts* *users in the test data who are interested in the topic.*

7.3 Personalizing Newsfeed in Social Media

In this section, we first present how we exploit the additional user posts, and then discuss how social network information can be integrated into a unified framework. Finally, we present the framework that utilizes both content and network information with its optimization.

7.3.1 Content Modeling with Multiple Instance Learning

Collaborative filtering models user interests by analyzing user-item correlations, which performs well when enough correlations are accumulated. However, a trending topic becomes popular immediately; so the correlations are not sufficient. Therefore, we aim to solve this problem by starting with a Content-Based Filtering (CBR) method. To predict a user's interests toward a trending topic based on content information, we adopt a logistic regression model, which has conventionally been used for CBR (Pazzani and Billsus, 2007; Agarwal and Chen, 2009). The formulation of the optimization problem is shown as follows,

$$f(\mathbf{u}_i) = \frac{1}{1 + \exp(-\mathbf{w} \cdot \psi(\mathbf{u}_i) - b)}, \tag{7.1}$$

where $f(\mathbf{u}_i)$ denotes the prediction result that whether user i is interested in the trending topic. b is the model bias and $\mathbf{w}$ is the vector of model parameters. b and $\mathbf{w}$ are the parameters to optimize in a logistic regression model. $\psi(\cdot)$ maps a user to an attribute vector.

$\psi(\cdot)$ generates an attribute vector based on posts of users. For a user with a positive label ($\mathbf{y}_i = 1$), posts explicitly containing the trending topic are very few. Therefore, if only these posts are used, the corresponding attribute vector should be very sparse. If all posts of the user are selected, noisy information would be unavoidably included. Therefore, an appealing model should be able to identify those implicitly correlated posts automatically. Motivated by the related research of computer vision, we propose to adopt Multi-Instance Learning (MIL).

A research problem in computer vision is lack of fine-grained labels. Take scene classification as an instance (Zhou and Zhang, 2006), although labels are usually only available for the entire picture, the key object that determines the label of a picture usually takes up a small portion of the entire picture. In order to better understand characteristics of a specific object of interest, in MIL, each picture is represented as a bag of subimages. Instead of learning the whole image, fractions that represent the object of interest are automatically identified and better modeled. Similar to MIL, in microblogging sites, a user contains a "bag" of posts and only few are related to a specific topic. Therefore, we pose the personalization problem into an MIL task by reformulating Eq. (7.1) as follows,

$$f(\mathbf{p}_{ik}) = \frac{1}{1 + \exp(-\mathbf{w} \cdot \mathbf{p}_{ik} - b)}, \tag{7.2}$$

where $f(\mathbf{p}_{ik})$ predicts the label for a single post $\mathbf{p}_{ik}$. By aggregating prediction of all posts, the estimation of a user can be obtained as follows,

$$f(\mathbf{u}_i) = \frac{\sum_{k=1}^{|\mathbf{u}_i|} f(\mathbf{p}_{ik}) \cdot \exp(\alpha f(\mathbf{p}_{ik}))}{\sum_{k=1}^{|\mathbf{u}_i|} \exp(\alpha f(\mathbf{p}_{ik}))}, \tag{7.3}$$

where a softmax function is the aggregate results of a user. α is a parameter introduced to determine the extent of softness of the combination. Given the label vector $\mathbf{y}$, the optimal parameters $\mathbf{w}, b$ for a topic j can be obtained through minimizing the following cost function,

$$\epsilon(\mathbf{w}, b) = \frac{1}{2} \sum_{i=1}^{m} (y_i - f(\mathbf{u}_i))^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} + \gamma |\mathbf{w}|_1, \tag{7.4}$$

where $\mathbf{w}^T\mathbf{w}$ is a regularization term to avoid over-fitting by penalizing the model complexity, of which the extent is controlled by λ. Since among the many words only few are correlated with a trending topic, we invoke an ℓ_1 regularizer to induce sparsity.

Microblogging users are interconnected by "following" and being "followed". According to the principle of homophily (McPherson *et al.*, 2001), interconnected friends are likely to have similar interests. In this section, we propose to leverage the homophily to alleviate the cold-start problem. We regard two users who follow each other as friends. Assume $\mathbf{E}$ represents friendship between users, where $\mathbf{e}_{it} = 1$ if $a_{it} = a_{ti} = 1$, and otherwise $\mathbf{e}_{it} = 0$. Therefore, the homophily can be modeled by minimizing the following graph-based regularizer

$$\sum_{e_{it} \in \mathbf{E}} e_{it}(f(\mathbf{u}_i) - f(\mathbf{u}_t))^2, \tag{7.5}$$

which smooths the prediction results of friends by penalizing the large difference between them. For the ease of integrating Eq. (7.5) with the optimization objective in Eq. (7.4), we introduce to rewrite the graph-based regularizer as a graph Laplacian form. Motivated by graph learning literature, the regularizer can be rewritten as $f^t\mathcal{L}f$, where $f \in \mathbb{R}^m$ is the prediction results of users. $\mathcal{L}$ is the normalized *Laplacian* matrix of the corresponding social graph (Merris, 1994) with the graph structure of $\mathbf{E}$. Specifically, the *Laplacian* $\mathbf{L}$ can be obtained through:

$$\mathbf{L} = \mathbf{D} - \mathbf{E},$$

where $\mathbf{D} \in \mathbb{R}^{m \times m}$ is a diagonal matrix and the diagonal elements are calculated as $d_{ii} = \sum_{k=1}^{m} e_{ik}$. The *normalized Laplacian* can then be calculated as:

$$\mathcal{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}.$$

Incorporating the normalized graph Laplacian norm as a regularizer rewrites the objective in Eq. (7.4) as follows:

$$\epsilon(\mathbf{w}, b) = \frac{1}{2}\sum_{i=1}^{m}(y_i - f(\mathbf{u}_i))^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} + \gamma|\mathbf{w}|_1 + \frac{\mu}{2}f^t\mathcal{L}f, \tag{7.6}$$

where the graph-based regularizer is reformulated and the resultant objective remains convex. μ controls the extent of penalization when the prediction results are different for friends. Since the amount of content information is massive, an efficient optimization method is required. Next, we introduce how we efficiently obtain optimal parameters $\mathbf{w}$, b with additional content and network information.

7.3.2 Model Fitting

For simplicity of presentation, we first augment $\mathbf{w}$ by incorporating b as $\mathbf{w}_0$, which can be implemented by adding an additional feature. Thus we aim to learn the optimal predictor as follows:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \frac{1}{2}\sum_{i=1}^{m}(y_i - f(\mathbf{u}_i))^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} + \gamma|\mathbf{w}|_1 + \frac{\mu}{2}f^t\mathcal{L}f. \tag{7.7}$$

Since we employ a logistic model independent for each feature, features can be calculated separately when updating $\mathbf{w}$. Since both the normalized Laplacian and ℓ_1-regularizer are convex, we adopt projected gradient descent to update each feature w_k as follows:

$$\begin{aligned}\frac{\partial\epsilon}{\partial w_k} &= \sum_{i=1}^{m}(y_i - f(\mathbf{u}_i))\frac{\partial f(\mathbf{u}_i)}{\partial w_k} + \lambda w_k + \gamma\cdot\mathrm{Sign}(w_k)\\ &+ \mu\sum_{ij=1}^{m}\mathcal{L}_{ji}f(\mathbf{u}_i)\frac{\partial f(\mathbf{u}_i)}{\partial w_k},\end{aligned} \tag{7.8}$$

where $\mathcal{L}_{ji}$ is the value of the corresponding entry in the normalized Laplacian matrix and $\frac{\partial f(\mathbf{u}_i)}{\partial w_k}$ is the gradient of softmax. The gradient of softmax can be further decomposed by each post $\mathbf{p}_{ij}$ as follows:

$$\frac{\partial f(\mathbf{u}_i)}{\partial w_k} = \sum_{j=1}^{|\mathbf{u}_i|} \frac{\partial f(\mathbf{u}_i)}{\partial f(\mathbf{p}_{ij})} \frac{\partial f(\mathbf{p}_{ij})}{\partial w_k}, \tag{7.9}$$

where the derivative of the logistic regression of posts can be computed by conventional approaches, and the derivative of the softmax aggregation function in terms of a post $\mathbf{p}_{ij}$ can be computed as follows:

$$\frac{\partial f(\mathbf{u}_i)}{\partial \mathbf{p}_{ij}} = \frac{(1 + \alpha f(\mathbf{p}_{ij}) - \alpha f(\mathbf{u}_i)) \exp(\alpha f(\mathbf{p}_{ij}))}{\sum_{j=1}^{|\mathbf{u}_i|} \exp(\alpha f(\mathbf{p}_{ij}))}. \tag{7.10}$$

In traditional MIL approaches, labels are only available for *bags*. In our case, labels are also available for some posts. Next, we will introduce the reason and present how we jointly model both of them.

7.3.3 Joint Modeling of Posts and Users

In order to integrate labels of posts, we propose to build up pseudo-users. It has been shown that directly incorporating instance labels would overshadow the effect of bag labels (Settles *et al.*, 2008). A possible way to avoid the problem is to create singleton bags. Motivated by related research in multi-instance learning (Settles *et al.*, 2008), we create a pseudo user for each labeled post, who contains only the labeled post. The nice property is to enable models to benefit from both user (bag)- and post (instance)-level training information. Adding pseudo users also results in a change of the graph. Here, we connect the pseudo user with its author only. The detailed algorithm is shown in Algorithm 6.

In Algorithm 6, the user set **U** and adjacency matrix **E** contain pseudo-users which are generated by labeled posts. Line 1 initializes the iteration identifier and other parameters, values of which are found through cross-validation on a holdout dataset. From line 4 to line 12 we aim to find the optimal learning rate with backtracking line search (Boyd and Vandenberghe, 2004). Line 13 updates the parameters with the

Algorithm 6 Personalizing Trending Topics in Microblogging

```
Input: Posts from Users: U;
    Vector of Twitter User Labels: y;
    Adjacency Matrix of Users: E;
    Parameters : $\lambda, \gamma, \mu$;
    Maximum Number of Iterations : $I$.
Output: Logisitic Predictors: w.
Initialize $t = 1$; Set $\lambda$, $\gamma$, $\mu$.
while Not convergent and $t \leq I$
    Calculate $\frac{\partial \epsilon}{\partial \mathbf{w}}$ with Eq.(7.8)
    Set $\tau = 1$
    loop
        If $\epsilon(\mathbf{w} - \tau\frac{\partial \epsilon}{\partial \mathbf{w}}) \geq \epsilon(\mathbf{w})$-$\frac{\tau}{2}||\frac{\partial \epsilon}{\partial w}||^2$ then
            $\tau = 0.5\tau$
        end if
        Else
            $\tau = 2 \times \tau$; Break
        end else
    end loop
    $\mathbf{w} = \mathbf{w} - \tau\frac{\partial \epsilon}{\partial w}$
end while
```

Table 7.1: Content-centric Features Used in This Study.

Feature Name	Explanation
Words	# of occurrences of words
Hashtags	# of occurrences of hashtags
BigramW	# of occurrences of bigrams of words
BigramH	# of occurrences of bigrams of hashtags
Emoticons	# of occurrences of emoticons
Sentiment	Avg. sentiment of emoticons

gradient.

7.3.4 Discussion

Next, we will introduce how the textual features are extracted from posts, and we will also discuss the time complexity of the proposed framework.

Features Used in this Study We derive six types of features in this section, which are shown in Table 7.1. Words directly characterize the content of posts. Hashtags are indicative for semantic of a post. We remove hashtags of trending topics. The bigram features of hashtags and words can represent common semantic in real applications. Also, we use the sentiment polarity of emoticons in posts as features. The sentiment of emoticons can be estimated through resolving the description of emoticons[5] with AFINN (Nielsen, 2011).

Time Complexity and Running Time The time complexity for an iteration over all users and features is $O(mnd^2)$, where m is the number of users, $n = max(|\mathbf{u}_i|)$ is the maximum number of posts, and d is the number of textual features. A nice property is that the computation can be employed in parallel. In particular, different features (w_k) can be updated simultaneously. The parallel implementation of the

[5] http://emojipedia.org/twitter/

Table 7.2: Statistics of the Dataset Used in This Study.

Topics	Users	Labeled Posts
1,012	101,351	10,151
Posts	Links	
2,015,802	20,046,715	

algorithm will be publicly available upon being published.

7.4 Experiments

In the experiments, we are applying our TTP approach on the real-world data obtained from Twitter. To collect for the dataset, we randomly collect 1,012 trending topics from Twitter, from June 6^{th}, 2016 to June 8^{th}, 2016 in the area of United States. In order to find potentially interested users, we randomly collect users who post during that period from Twitter's Streaming API[6]. For each user, we obtain their followers and friends to build up the adjacency matrix and collect up to 20 most recent posts. Statistics on the dataset are shown in Table 7.2. In order to train the model, we use the posts that are generated within the first hour when the topic starts trending as training data. In order to test the model, we use users who post on the trending topic after the first hour.

There are three positive parameters involved in the experiments, including λ, γ, and μ in Eq.(7.7). λ is to control overfitting and makes the learned model more robust. γ is to control the sparsity of the learned model. μ is to control the contribution of network information. As a common practice, all the parameters can be tuned through cross-validation. We set $\lambda = 0.1$, $\gamma = 0.1$, and $\mu = 0.1$, though in our experience the parameters do not significantly impact performance.

We follow standard personalization settings to evaluate the performance. In par-

[6] https://dev.twitter.com/streaming/reference/post/statuses/filter

ticular, since the softmax output is a real number, we use the metric of "root-mean-square-root" (RMSE) on different methods with the Twitter dataset. RMSE can be calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{m}\sum_{j=1}^{t}(\mathbf{y}_{ij} - f_j(\mathbf{u}_i))^2}{m \times t}}, \tag{7.11}$$

where m is the number of users and t is the number of trending topics. $\mathbf{y}_{ij}$ is the ground truth of user i on topic j. $f_j(\cdot)$ is the prediction function for topic j. The difference denotes the distance between the prediction and the ground truth. $m \times t$ is the total number of predictions and it normalizes the results. In our settings, the RMSE indicates the difference between the prediction and users' true interests towards the trending topics. The smaller RMSE represents a better performance.

7.4.1 Performance Comparisons

In order to answer the first question about the personalization effectiveness, in addition to TTP, we compare with the following state-of-the-art personalization methods:

- *PMF*: Collaborative filtering has been regarded as a state-of-the-art recommendation method in various areas such as movies. In this section, we adopt BPMF (Salakhutdinov and Mnih, 2008), which adopts fully Bayesian treatment of the Probabilistic Matrix Factorization. It is considered to be one of the most effective methods when the training data is sparse.

- *CBF* and *CBF+*: As discussed in the survey (Veltkamp *et al.*, 2013), the most effective content-based approach for personalization is to calculate the similarity based on both the attribute vector of keywords. In this section, we collect posts containing the trending topic hashtag in the training set to represent a trending topic, and user posts to represent a user. We adopt the commonly-used TF-IDF to weight word features, and the corresponding similarity is used for personalization. In CBF, we use posts hashtagged with the trending topic to represent a user. While in CBF+, all posts of an interested user are used. CBF and CBF+ are adopted to compare TTP with effective content-based filtering methods.

- *LMGR* and *LMGR+*: The proposed TTP jointly exploits content and social network information. In order to investigate the effectiveness of TTP, we compare with state-of-the-art methods that utilize both content and network information. LMGR accurately predicts labels of web documents (Zhang *et al.*, 2006) by simultaneously modeling content and hyperlinks. Similarly, LMGR exploits only the posts hashtagged by the trending topic while LMGR+ uses all posts.

- *SocDim*: Social interactions have been regarded as an effective data source for determining user interests. In this section, we adopt SocDim (Tang and Liu, 2009), which learns user interests by projecting social relations into a low dimensional space. SocDim has commonly been used for categorizing users according to interests, and can be considered as a state-of-the-art method for relational learning on social networks.

- *Random*: Because there are much more negative training examples than the positive examples in the dataset, the absolute value of RMSE is not very meaningful. For comparison purposes, we also use a Random baseline that uniformly selects trending topics for each user.

The comparison is shown in Figure 7.2. Since there are much more negative training data instances than the positive ones in the Twitter dataset, the absolute value

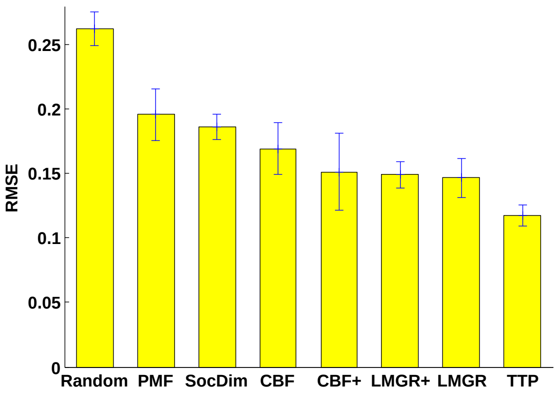


Figure 7.2: Performance Comparisons for Different Personalization Methods with 95% Confidence Interval.

of RMSE is not very meaningful. The RMSE of Random baseline is 0.2621. We are expecting the relative decreasing of RMSE for a more effective approach. Based on results shown in Figure 7.2, we draw following observations. Traditional recommendation approaches, *i.e.*, content-based (CBF, CBF+) and collaborative filtering (PMF) cannot effectively personalize trending topics. SocDim outperforms PMF and Random, showing that knowledge that is useful for identifying user interests exists in the network structures. The proposed TTP outperforms existing methods that directly integrate social network structures with user contents (LMGR, LMGR+) by selecting the related content from the massive amount of historical information, instead of ignoring or adopting all. The result demonstrates that TTP is effective in inspecting user interests and personalizing trending topics.

7.4.2 Earliness of Personalization

A key objective of our study is to find interesting trending topics at an early stage before they become obsolete. More user posts and other data are available for training at a later stage with the topic being trending. However, late recommendations are much less practically useful, since a topic trended yesterday may get outdated and less interesting to read. Therefore, we investigate how effective TTP is when less

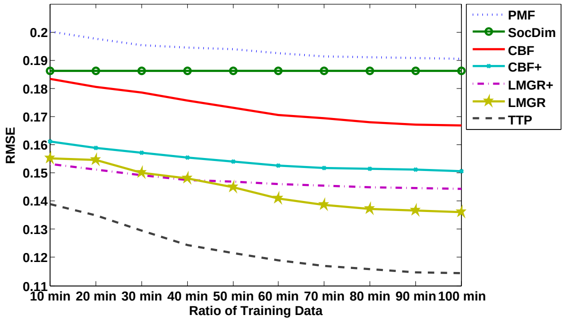


Figure 7.3: Performance of Different Models with Chronologically Additional Training Data, While Socdim Uses the Social Network Structures.

training data is available during the early period.

The results are shown in Figure 7.3. In order to evaluate earliness, we train models by additionally using training data by its chronological order. In particular, we additionally add training data based on the time order they were generated after the trending topic started trending. Since SocDim only uses the network information, the performance is constant. The RMSE of other methods decreases with more training data being added. According to the results, the best baseline, LMGR, achieves RMSE of TTP with training data of first ten minutes by a lag of 90 minutes. Therefore, the empirical results show that the use of TTP not only yields low error rate, but also finds interesting trending topics hours faster than traditional personalization approaches.

7.5 Summary

Trending topics, which are immediately popular topics on social media sites, have been popular among social media users as an information source. With the fast increase of trending topics, it becomes a crucial task for microblogging sites to help social media users find the topic they are interested in. Therefore, in this section, we propose the Trending Topic Personalization approach (TTP) to personalize trending topics at an early stage. TTP tackles the cold-start problem through jointly exploiting social media posts and social interactions between users. Through experiments on real-world data, we have demonstrated the gains of performance and earliness of the proposed TTP over other state-of-the-art approaches.

Chapter 8

CONCLUSION AND FUTURE WORK

In this chapter, we conclude the dissertation by summarizing the contributions and highlighting the potential research directions in the future.

8.1 Conclusion

As witnessed in recent incidents of misinformation, social media platforms have allowed for rumors and fake news to spread to a large group of people rapidly. Misinformation could cause catastrophic effects in the real world within a short period, such as driving a wedge between people, and arousing astonishment and anxiety among people. Detecting misinformation in social media is of considerable significance to maintain the quality of user experiences, and to improve the conversational health of digital communities. Detecting misinformation can be very challenging because of the adversarial attacks, manipulated content, and rapid growth at the early stage. Through tackling the challenges of misinformation in social media, the contributions of the dissertation can be summarized from two perspectives. First, I formalize the novel problem of misinformation detection in social media and the computational challenges. Second, I am able to propose effective algorithms to tackle the challenges and mitigate the negative effect of misinformation. In conclusion, I investigate different aspects of the problem to characterize and detect misinformation in social media.

I study adversarial attacks of misinformation spreaders that present novel challenges. In particular, I investigate how the camouflage of misinformation spreaders can be identified with label information only for accounts. The proposed method utilizes discriminant analysis to discover the key post that distinguishes misinformation spreaders from legitimate users. Also, I present an efficient algorithm to solve the proposed non-smooth convex optimization problem. Experimental results on real-world Twitter datasets demonstrate that the proposed framework can effectively utilize available information to outperform the state-of-the-art approaches.

Content information provides limited information for the problem of misinformation detection. Motivated by the fact that informative patterns exist behind the diffusion of misinformation and contextual information is pervasively available, I focus on extracting descriptive from contextual information to facilitate the task. In particular, I propose a novel method that classifies social media messages with diffusion traces in social networks. To deal with the trace data, we introduce an end-to-end classification model based on LSTM-RNNs. In order to alleviate the data sparsity, I propose an embedding method that captures both social proximity and community structures. Experimental results with real-world datasets show that the proposed method effectively classifies social media messages and is especially useful when content information is insufficient.

Misinformation in social media grows and evolves rapidly. After it goes viral, it is extremely difficult to eliminate their existence. Therefore, I propose to directly train a classifier based on readily available labeled data from prior incidents in order to detect misinformation at an early stage. Motivated by traditional studies on misinformation, I introduce a novel framework that jointly clusters data, selects features, and trains classifiers. An optimization approach is also presented to solve the problem efficiently. The proposed framework largely breaks the bottleneck of the time lag from annotating datasets. Experimental results illustrate the effectiveness and earliness of the proposed method on real-world data.

Many studies nowadays rely on social media as their data source. Therefore, I propose to mitigate the negative effect of misinformation for two common machine learning tasks, relational learning, and social personalization. Relational learning focuses on classifying and tagging social media users, and social personalization aims to find interesting content for a user in an unsupervised manner. For the first task, I propose a unified framework that simultaneously selects data instances and learn a relational learning model. In order to allow for efficient optimization, we utilize the social community structure to effectively find groups of instances. I also transform the combinatorial problem into a convex optimization problem with relaxations. Experimental results on real-world datasets show the superiority of the proposed approach over competitive baseline methods. For the second task, I propose a novel approach to personalize content in social media at an early stage. The proposed method solves the cold-start problem through jointly exploiting social media posts and social interactions between users. Through experiments on real-world data, I have demonstrated the gains of performance and earliness of the proposed method over other state-of-the-art approaches.

8.2 Future Work

(1) How to predict the potential influence of misinformation in social media?

As an instance of classification, existing misinformation detection methods focus on optimizing classification accuracy. In real-world applications, however, detecting an influential spreader is may be more useful than ten unimportant ones that can hardly spread misinformation to regular users. It will be interesting to define influence of misinformation spreaders and formulate a computational problem to cope with it.

(2) How are misinformation spreaders spreading misinformation and attracting attention? Existing research mostly focuses on the spreaders - or the accounts that post misinformation in social media. In the real world, a spreader would more than that to “spread” misinformation, such as commenting under certain topics, making friends with similar communities, and even privately messaging interested accounts. In addition to detecting them, it would be interesting to discover and understand such spreading behaviors, which may ultimately facilitate building a robust detection system.

(3) How to make detection methods robust to adversarial attacks, or how to exploit adversarial learning to enhance a detection method? Adversarial machine learning aims to enable machine learning methods to be robust and effective in the presence of adversarial attacks. Current research focuses on adversarial attacks of misinformation spreaders, however, if there is a malicious adversary that has partial or full knowledge of the misinformation detection algorithm, existing methods can be vulnerable. It will be interesting to discover robust methods in the presence of adversarial attacks.

REFERENCES

Acemoglu, D., A. Ozdaglar and A. ParandehGheibi, "Spread of (mis) information in social networks", Games and Economic Behavior **70**, 2, 194–227 (2010).

Agarwal, D. and B.-C. Chen, "Regression-based latent factor models", in "Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 19–28 (ACM, 2009).

Agrawal, D., C. Budak and A. El Abbadi, "Information diffusion in social networks: observing and affecting what society cares about", in "Proceedings of CIKM", pp. 2609–2610 (2011).

Allport, G. W. and L. Postman, "The basic psychology of rumor", Transactions of the New York Academy of Sciences (1945).

Allport, G. W. and L. Postman, "The psychology of rumor.", (1947).

Anthony, S., "Anxiety and rumor", The Journal of Social Psychology (1973).

Armijo, L., "Minimization of functions having lipschitz continuous first partial derivatives", Pacific Journal of mathematics **16**, 1, 1–3 (1966).

Barbier, G., Z. Feng, P. Gundecha and H. Liu, "Provenance data in social media", Synthesis Lectures on Data Mining and Knowledge Discovery **4**, 1, 1–84 (2013).

Benevenuto, F., G. Magno, T. Rodrigues and V. Almeida, "Detecting spammers on twitter", in "Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)", vol. 6, p. 12 (2010).

Blondel, V. D., J.-L. Guillaume, R. Lambiotte and E. Lefebvre, "Fast unfolding of communities in large networks", Journal of Statistical Mechanics: Theory and Experiment **2008**, 10, P10008 (2008).

Bobadilla, J., F. Ortega, A. Hernando and A. Gutiérrez, "Recommender systems survey", Knowledge-Based Systems **46**, 109–132 (2013).

Bollen, J., H. Mao and X. Zeng, "Twitter mood predicts the stock market", Journal of computational science **2**, 1, 1–8 (2011).

Bottou, L., "Large-scale machine learning with stochastic gradient descent", in "Proceedings of COMPSTAT'2010", pp. 177–186 (Springer, 2010).

Box, M., W. H. Swann and D. Davies, "Non-linear optimization techniques", (1969).

Boyd, S. and L. Vandenberghe, *Convex optimization* (Cambridge university press, 2004).

Cai, D., X. He, J. Han and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation", IEEE Transactions on Pattern Analysis and Machine Intelligence **33**, 8, 1548–1560 (2011).

Cao, N., C. Shi, S. Lin, J. Lu, Y.-R. Lin and C.-Y. Lin, “Targetvue: Visual analysis of anomalous user behaviors in online communication systems”, IEEE transactions on visualization and computer graphics **22**, 1, 280–289 (2016).

Centola, D., “The spread of behavior in an online social network experiment”, science **329**, 5996, 1194–1197 (2010).

Chen, T. and C. Guestrin, “Xgboost: A scalable tree boosting system”, in “International Conference on Knowledge Discovery and Data Mining”, pp. 785–794 (ACM, 2016).

Cho, K., B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation”, arXiv preprint arXiv:1406.1078 (2014).

Clarke, L., “On cayley’s formula for counting trees”, Journal of the London Mathematical Society **1**, 4, 471–474 (1958).

Del Vicario, M., A. Bessi, F. Zollo, F. Petroni, A. Scala, G. Caldarelli, H. E. Stanley and W. Quattrociocchi, “The spreading of misinformation online”, Proceedings of the National Academy of Sciences **113**, 3, 554–559 (2016).

DiFonzo, N. and P. Bordia, *Rumor Psychology: Social and Organizational Approaches* (American Psychological Association, 2007).

Donoho, D. L. and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization”, PNAS **100**, 5, 2197–2202 (2003).

Du, L., X. Li and Y.-D. Shen, “Robust nonnegative matrix factorization via half-quadratic minimization”, in “Data Mining (ICDM), 2012 IEEE 12th International Conference on”, pp. 201–210 (IEEE, 2012).

Eldardiry, H. and J. Neville, “An analysis of how ensembles of collective classifiers improve predictions in graphs”, in “Proceedings of the 21st CIKM”, pp. 225–234 (ACM, 2012).

Fei, G., A. Mukherjee, B. Liu, M. Hsu, M. Castellanos and R. Ghosh, “Exploiting burstiness in reviews for review spammer detection.”, ICWSM **13**, 175–184 (2015).

Fletcher, R., *Practical Methods of Optimization* (John Wiley & Sons, 2013).

Fukunaga, K., *Introduction to statistical pattern recognition* (Academic press, 2013).

Gomez Rodriguez, M., J. Leskovec and A. Krause, “Inferring networks of diffusion and influence”, in “Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining”, pp. 1019–1028 (ACM, 2010).

Goodfellow, I., Y. Bengio and A. Courville, *Deep learning* (MIT Press, 2016).

Gu, G., J. Zhang and W. Lee, “Botsniffer: Detecting botnet command and control channels in network traffic”, (2008).

Guy, I., N. Zwerdling, I. Ronen, D. Carmel and E. Uziel, “Social media recommendation based on people and tags”, in “Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval”, pp. 194–201 (ACM, 2010).

Hallac, D., J. Leskovec and S. Boyd, “Network lasso: Clustering and optimization in large graphs”, in “Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining”, pp. 387–396 (ACM, 2015).

Han, J. S. and B. J. Park, “Efficient detection of content polluters in social networks”, in “IT Convergence and Security 2012”, pp. 991–996 (Springer, 2013).

Hastie, T., R. Tibshirani and M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations* (CRC Press, 2015).

Hawkins, S. and et al., “Outlier detection using replicator neural networks”, in “Data Warehousing and Knowledge Discovery”, pp. 170–180 (2002).

Herlocker, J. L., J. A. Konstan and J. Riedl, “Explaining collaborative filtering recommendations”, in “Proceedings of the 2000 ACM conference on Computer supported cooperative work”, pp. 241–250 (ACM, 2000).

Hogg, M. A., “Social identity theory”, in “Understanding Peace and Conflict Through Social Identity Theory”, pp. 3–17 (Springer, 2016).

Hooi, B., H. A. Song, A. Beutel, N. Shah, K. Shin and C. Faloutsos, “Fraudar: bounding graph fraud in the face of camouflage”, in “Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)”, pp. 895–904 (2016).

Hu, X., J. Tang and H. Liu, “Leveraging knowledge across media for spammer detection in microblogging”, in “37th International ACM SIGIR conference”, pp. 547–556 (ACM, 2014).

Hu, X., J. Tang, Y. Zhang and H. Liu, “Social spammer detection in microblogging.”, in “IJCAI”, vol. 13, pp. 2633–2639 (2013).

Jensen, D., J. Neville and B. Gallagher, “Why collective inference improves relational classification”, in “KDD”, pp. 593–598 (ACM, 2004).

Ji, S. and J. Ye, “An accelerated gradient method for trace norm minimization”, in “ICML”, pp. 457–464 (ACM, 2009).

Jiang, M., P. Cui and C. Faloutsos, “Suspicious behavior detection: Current trends and future directions”, Intelligent Systems **31**, 1, 31–39 (2016).

Jindal, N. and B. Liu, “Review spam detection”, in “Proceedings of the 16th international conference on World Wide Web”, pp. 1189–1190 (ACM, 2007).

Joachims, T., “Text categorization with support vector machines: Learning with many relevant features”, in “European conference on machine learning”, pp. 137–142 (Springer, 1998).

Karlova, N. and K. Fisher, “Plz rt: A social diffusion model of misinformation and disinformation for understanding human information behaviour”, Information Research **18**, 1 (2013).

Kempe, D., J. Kleinberg and É. Tardos, “Maximizing the spread of influence through a social network”, in “Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining”, pp. 137–146 (ACM, 2003).

Kermack, W. O. and A. G. McKendrick, “A contribution to the mathematical theory of epidemics”, in “Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences”, vol. 115, pp. 700–721 (1927).

Kietzmann, J. H., K. Hermkens, I. P. McCarthy and B. S. Silvestre, “Social media? get serious! understanding the functional building blocks of social media”, Business horizons (2011).

Kim, Y., “Convolutional neural networks for sentence classification”, arXiv preprint arXiv:1408.5882 (2014).

Kolari, P., T. Finin and A. Joshi, “Svms for the blogosphere: Blog identification and splog detection.”, in “AAAI Spring Symposium”, pp. 92–99 (2006).

Kong, D., R. Fujimaki, J. Liu, F. Nie and C. Ding, “Exclusive feature learning on arbitrary structures via l12-norm”, in “NIPS”, pp. 1655–1663 (2014).

Kumar, S., F. Morstatter and H. Liu, *Twitter data analytics* (Springer, 2014).

Laumann, E. O. and F. U. Pappi, *Networks of collective action: A perspective on community influence systems* (Elsevier, 2013).

Lawson, C. L. and R. J. Hanson, *Solving least squares problems*, vol. 161 (SIAM, 1974).

Lee, D. D. and H. S. Seung, “Algorithms for non-negative matrix factorization”, in “Advances in neural information processing systems”, pp. 556–562 (2001).

Lee, K., J. Caverlee and S. Webb, “The social honeypot project: protecting online communities from spammers”, in “Proceedings of the 19th international conference on World wide web”, pp. 1139–1140 (ACM, 2010a).

Lee, K., J. Caverlee and S. Webb, “Uncovering social spammers: social honeypots+ machine learning”, in “Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval”, pp. 435–442 (ACM, 2010b).

Lee, K., B. D. Eoff and J. Caverlee, “Seven months with the devils: A long-term study of content polluters on twitter.”, in “ICWSM”, (Citeseer, 2014).

Lemaréchal, C. and C. Sagastizábal, “Practical aspects of the moreau–yosida regularization: Theoretical preliminaries”, SIAM Journal on Optimization **7**, 2, 367–385 (1997).

Li, J., X. Hu, L. Wu and H. Liu, "Robust unsupervised feature selection on networked data", in "2016 SIAM International Conference on Data Mining (SDM)", pp. 387–395 (2016).

Liu, B., "Sentiment analysis and opinion mining", Synthesis lectures on human language technologies **5**, 1, 1–167 (2012).

Liu, J., S. Ji and J. Ye, "Multi-task feature learning via efficient l 2, 1-norm minimization", in "UAI", pp. 339–348 (2009).

Liu, J. and J. Ye, "Moreau-Yosida regularization for grouped tree structure learning", in "NIPS", pp. 1459–1467 (2010).

Manning, C. D., M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard and D. McClosky, "The Stanford CoreNLP natural language processing toolkit", in "Association for Computational Linguistics (ACL) System Demonstrations", pp. 55–60 (2014), URL `http://www.aclweb.org/anthology/P/P14/P14-5010`.

Markines, B., C. Cattuto and F. Menczer, "Social spam detection", in "International Workshop on Adversarial Information Retrieval on the Web", (ACM, 2009).

Mccord, M. and M. Chuah, "Spam detection on twitter using traditional classifiers", in "Autonomic and trusted computing", pp. 175–186 (Springer, 2011).

McKelvey, K. R. and F. Menczer, "Truthy: Enabling the study of online social networks", in "CSCW", pp. 23–26 (ACM, 2013).

McPherson, M., L. Smith-Lovin and J. M. Cook, "Birds of a feather: Homophily in social networks", Annual review of sociology pp. 415–444 (2001).

Meier, L., S. Van De Geer and P. Bühlmann, "The group lasso for logistic regression", Journal of the Royal Statistical Society: Series B (Statistical Methodology) pp. 53–71 (2008).

Merris, R., "Laplacian matrices of graphs: a survey", Linear algebra and its applications **197**, 143–176 (1994).

Moré, J. J. and D. J. Thuente, "Line search algorithms with guaranteed sufficient decrease", ACM Transactions on Mathematical Software (TOMS) **20**, 3, 286–307 (1994).

Mukherjee, A. and et al., "Spotting opinion spammers using behavioral footprints", in "KDD", pp. 632–640 (ACM, 2015).

Neville, J. and D. Jensen, "Leveraging relational autocorrelation with latent group models", in "Proceedings of the 4th international workshop on Multi-relational mining", pp. 49–55 (ACM, 2005).

Ng, A. Y., "Feature selection, l 1 vs. l 2 regularization, and rotational invariance", in "ICML", p. 78 (2004).

Nielsen, F., *AFINN* (Informatics and Mathematical Modelling, Technical University of Denmark, 2011).

Niu, Y., H. Chen, F. Hsu, Y.-M. Wang and M. Ma, “A quantitative study of forum spamming using context-based analysis.”, in “NDSS”, (2007).

Nocedal, J. and S. Wright, *Numerical optimization* (Springer Science & Business Media, 2006).

Obozinski, G., B. Taskar and M. Jordan, “Multi-task feature selection”, Statistics Department, UC Berkeley, Tech. Rep **2** (2006).

Oh, O., M. Agrawal and H. R. Rao, “Community intelligence and social media services: A rumor theoretic analysis of tweets during social crises.”, Mis Quarterly **37**, 2, 407–426 (2013).

Palangi, H., L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song and R. Ward, “Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval”, IEEE/ACM Transactions on Audio, Speech and Language Processing **24**, 4, 694–707 (2016).

Pariser, E., *The filter bubble: What the Internet is hiding from you* (Penguin UK, 2011).

Pazzani, M. J. and D. Billsus, “Content-based recommendation systems”, in “The adaptive web”, pp. 325–341 (Springer, 2007).

Perozzi, B., R. Al-Rfou and S. Skiena, “Deepwalk: Online learning of social representations”, in “International Conference on Knowledge Discovery and Data Mining”, pp. 701–710 (ACM, 2014).

Piper, P. S., “Better read that again: Web hoaxes & misinformation”, (2001).

Qazvinian, V., E. Rosengren, D. R. Radev and Q. Mei, “Rumor has it: Identifying misinformation in microblogs”, in “Proceedings of the Conference on Empirical Methods in Natural Language Processing”, pp. 1589–1599 (Association for Computational Linguistics, 2011).

Ratkiewicz, J., M. D. Conover, M. Meiss, B. Gonçalves, A. Flammini and F. M. Menczer, “Detecting and tracking political abuse in social media”, in “Fifth international AAAI conference on weblogs and social media”, (2011).

Rennie, J. D., “Smooth hinge classification”, Proceeding of Massachusetts Institute of Technology (2005).

Rosnow, R. L., “Inside rumor: A personal journey.”, American Psychologist **46**, 5, 484 (1991).

Salakhutdinov, R. and A. Mnih, “Bayesian probabilistic matrix factorization using markov chain monte carlo”, in “Proceedings of the 25th international conference on Machine learning”, pp. 880–887 (ACM, 2008).

Sampson, J., F. Morstatter, L. Wu and H. Liu, “Leveraging the implicit structure within social media for emergent rumor detection”, Conference on Information and Knowledge Management (2016a).

Sampson, J., F. Morstatter, L. Wu and H. Liu, “Leveraging the implicit structure within social media for emergent rumor detection”, in “Proceedings of the 25th ACM International on Conference on Information and Knowledge Management”, pp. 2377–2382 (ACM, 2016b).

Settles, B., M. Craven and S. Ray, “Multiple-instance active learning”, in “Advances in neural information processing systems”, pp. 1289–1296 (2008).

Shao, C., G. L. Ciampaglia, A. Flammini and F. Menczer, “Hoaxy: A platform for tracking online misinformation”, in “WWW”, (2016).

Sherman, J. and W. J. Morrison, “Adjustment of an inverse matrix corresponding to a change in one element of a given matrix”, The Annals of Mathematical Statistics **21**, 1, 124–127 (1950).

Singh, A. P. and G. J. Gordon, “Relational learning via collective matrix factorization”, in “ACM SIGKDD International Conference on Knowledge Discovery and Data Mining”, pp. 650–658 (2008).

Socher, R., A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Potts *et al.*, “Recursive deep models for semantic compositionality over a sentiment treebank”, in “Proceedings of the conference on empirical methods in natural language processing (EMNLP)”, vol. 1631, p. 1642 (Citeseer, 2013).

Tang, J. and H. Liu, “Feature selection with linked data in social media”, (2012).

Tang, J., M. Qu, M. Wang, M. Zhang, J. Yan and Q. Mei, “Line: Large-scale information network embedding”, in “Proceedings of the 24th International Conference on World Wide Web”, pp. 1067–1077 (ACM, 2015).

Tang, L. and H. Liu, “Relational learning via latent social dimensions”, in “Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining”, pp. 817–826 (ACM, 2009).

Tax, C. V., D.M.J., “MIL, a Matlab toolbox for multiple instance learning”, URL `http://prlab.tudelft.nl/david-tax/mil.html`, version 1.1.0 (2015).

Thomas, K., C. Grier, D. Song and V. Paxson, “Suspended accounts in retrospect: an analysis of twitter spam”, in “IMC”, pp. 243–258 (ACM, 2011).

Thonnard, O. and M. Dacier, “A strategic analysis of spam botnets operations”, in “Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference”, pp. 162–171 (ACM, 2011).

Tucker, C. E., “Social networks, personalized advertising, and privacy controls”, (American Marketing Association, 2014).

Tumasjan, A., T. O. Sprenger, P. G. Sandner and I. M. Welpe, "Predicting elections with twitter: What 140 characters reveal about political sentiment.", Icwsm **10**, 1, 178–185 (2010).

Veltkamp, R., H. Burkhardt and H.-P. Kriegel, *State-of-the-art in content-based image and video retrieval*, vol. 22 (Springer Science & Business Media, 2013).

Von Luxburg, U., "A tutorial on spectral clustering", Statistics and computing **17**, 4, 395–416 (2007).

Wang, G., S. Xie, B. Liu and P. S. Yu, "Review graph based online store review spammer detection", in "ICDM", pp. 1242–1247 (IEEE, 2011).

Wang, G., S. Xie, B. Liu and P. S. Yu, "Identify online store review spammers via social review graph", TIST **3**, 4, 61 (2012).

Wang, S., J. Tang and H. Liu, "Embedded unsupervised feature selection.", in "AAAI", pp. 470–476 (2015).

Webb, S., J. Caverlee and C. Pu, "Social honeypots: Making friends with a spammer near you.", in "CEAS", (2008).

Wen, Z. and W. Yin, "A feasible method for optimization with orthogonality constraints", Mathematical Programming **142**, 1-2, 397–434 (2013).

Wu, L., X. Hu and H. Liu, "Relational learning with social status analysis", in "Proceedings of the Ninth ACM International Conference on Web Search and Data Mining", pp. 513–522 (ACM, 2016a).

Wu, L., X. Hu, F. Morstatter and H. Liu, "Detecting camouflaged content polluters.", in "ICWSM", pp. 696–699 (2017a).

Wu, L., J. Li, X. Hu and H. Liu, "Gleaning wisdom from the past: Early detection of emerging rumors in social media", in "Proceedings of the 2017 SIAM International Conference on Data Mining", pp. 99–107 (SIAM, 2017b).

Wu, L., F. Morstatter, X. Hu and H. Liu, "Mining misinformation in social media", in "Big Data in Complex and Social Networks", pp. 123–152 (CRC Press, 2016b).

Xu, Z., V. Tresp, S. Yu and K. Yu, "Nonparametric relational learning for social network analysis", in "KDD 2008 Workshop on Social Network Mining and Analysis", (2008).

Yang, C., R. Harkreader, J. Zhang, S. Shin and G. Gu, "Analyzing spammers' social networks for fun and profit: a case study of cyber criminal ecosystem on twitter", in "Proceedings of the 21st international conference on World Wide Web", pp. 71–80 (ACM, 2012).

Yang, C., Z. Liu, D. Zhao, M. Sun and E. Y. Chang, "Network representation learning with rich text information.", in "IJCAI", pp. 2111–2117 (2015).

Yang, J., J. McAuley and J. Leskovec, “Community detection in networks with node attributes”, in “Data Mining (ICDM), 2013 IEEE 13th international conference on”, pp. 1151–1156 (IEEE, 2013).

Zhang, T., A. Popescul and B. Dom, “Linear prediction models with graph regularization for web-page categorization”, in “Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining”, pp. 821–826 (ACM, 2006).

Zhao, J., N. Cao, Z. Wen, Y. Song, Y.-R. Lin and C. Collins, “# fluxflow: Visual analysis of anomalous information spreading on social media”, IEEE Transactions on Visualization and Computer Graphics **20**, 12, 1773–1782 (2014).

Zhao, Z., P. Resnick and Q. Mei, “Enquiring minds: Early detection of rumors in social media from enquiry posts”, in “WWW”, pp. 1395–1405 (2015).

Zhou, Z.-H., “Multi-instance learning: A survey”, Technical Report,AI Lab, Nanjing University (2004).

Zhou, Z.-H. and M.-L. Zhang, “Multi-instance multi-label learning with application to scene classification”, in “Advances in neural information processing systems”, pp. 1609–1616 (2006).

Zinkevich, M., M. Weimer, L. Li and A. J. Smola, “Parallelized stochastic gradient descent”, in “Advances in neural information processing systems”, pp. 2595–2603 (2010).

Biographical Sketch

Liang Wu has been a PhD student of Computer Science and Engineering at Arizona State University since August, 2014. The focus of his research is in the areas of misinformation and content polluter detection, and statistical relational learning. He has published over 20 innovative works in major international conferences in data mining and information retrieval, such as SIGIR, WSDM, ICDM, SDM, ICWSM, CIKM and AAAI. Liang has won the Honorable Mention Award of KDD Cup 2012, ranking 3rd on the leaderboard. He is also an author of 2 patents and 2 book chapters, and he is a tutorial speaker at SBP'16 and ICDM'17. He has been an intern at Nokia Research Center, Microsoft Research Asia, NEC Labs China, IBM China Research Labs, Etsy and Airbnb. He obtained his master's degree from Chinese Academy of Sciences in 2014 and bachelor's from Beijing Univ. of Posts and Telecom., China in 2011. More information can be found at http://www.public.asu.edu/˜liangwu1/.