A Probabilistic Model for Malicious User and Rumor Detection on Social Media

Yihong Zhang
Department of Multimedia Engineering
Osaka University
Osaka, Japan
yhzhang7@gmail.com

Takahiro Hara
Department of Multimedia Engineering
Osaka University
Osaka, Japan
hara@ist.osaka-u.ac.jp

Abstract

Rumor detection in recent years has emerged as an important research topic, as fake news on social media now has more significant impacts on people's lives, especially during complex and controversial events. Most existing rumor detection techniques, however, only provide shallow analyses of users who propagate rumors. In this paper, we propose a probabilistic model that describes user maliciousness with a two-sided perception of rumors and true stories. We model not only the behavior of retweeting rumors, but also the intention. We propose learning algorithms for discovering latent attributes and detecting rumors based on such attributes, supposedly more effectively when the stories involve retweets with mixed intentions. Using real-world rumor datasets, we show that our approach can outperform existing methods in detecting rumors, especially for more confusing stories. We also show that our approach can capture malicious users more effectively.

1. Introduction

Social media such as Twitter and Weibo have increasingly become important platforms for getting information and exchanging ideas. Statistics show that Twitter has more than 68 million monthly active users in U.S, while 36 percent of Americans aged 18 to 29 years old use Twitter [1]. These include some very influential users who are opinion leaders and have millions of followers [1]. During controversial events such as presidential elections, social media platforms are treated as arenas for sharing sentiments and exchanging attacks [2]. Messages propagate on some social media platforms such as Twitter and Weibo through an activity called $retweeting$, with which a user can effortlessly repeat a message for all their followers to see. Through retweeting, messages on social media have the potential to reach hundreds or thousands of people in a short frame of time [3]. What is more, discussions on social media have been shown to have significant impact on the real world. It has been shown that election results are highly correlated with and can be predicted from social media discussions [4].

Given this situation, we can say that information circulating on social media can have significant impacts on people's lives. And harmful information such as fake news can also have a strong consequence, such as causing public fear during crises [5]. Thus in recent years, a large number of researches have been done on social media rumors, including tracking rumor propagation over time, user roles, and signals for automatic rumor detection [6, 7, 8, 5]. In some works, a rumor is defined as a story whose veracity is not determined [8, 9]. In other works, a rumor is simply a fake news or misinformation [6, 7]. In this work, we follow the latter approach, and consider a story either as a fake story, called a rumor, or a true story.

Different from a number of works that deal with perceptual credibility [10, 11], in this work we deal with story veracity, i.e., whether the story contained in a message is true or false as verified in the real world. As a unique characteristic, our work is centered on two rumor-related latent attributes of users who involve in the message propagation, namely, *malicious sensitivity* and *malicious perplexity*. Malicious sensitivity indicates how likely a user would retweet rumor, knowing that it is a rumor, and malicious perplexity indicates how likely a user would retweet a true story, mistaking it for a rumor. Such attributes can be used in turn to detect rumors. Our rumor detection method is based on the users who retweeted the story, without examining its content. This gives us more flexibility on the type of stories that can be handled. For example, picture messages and non-English messages, which many existing approaches based on text and linguistic features cannot handle [12, 13], can be handled by our approach. We purposefully leave out the content of the story in order to focus on the user and the propagation. However, story contents

[1] https://www.omnicoreagency.com/twitter-statistics/

URI: https://hdl.handle.net/10125/64051
978-0-9981331-3-3

can be added in further implementations to improve detection accuracy.

The goal of our approach is to discover these latent user attributes as well as to detect story veracity using such attributes. To the best of our knowledge, our work is the first study of learning user maliciousness and story veracity labels in the same framework. We expect our work to have critical practical implications. We have seen nowadays an increasing number of rumors that are circulating on social media, and it has become desirable to not only detect rumors at an early stage, but also impose restriction and even punish malicious users. The findings of our work can be applied to make such decisions more effective. Furthermore, we consider that our approach has a generality that can be applied to other social media analyses where the label of a post is dependent on the type of retweeting users. One example can be detecting posts that have healing effects and users who suffer depression.

To be presented in detail in Section 5, our approach is built on a latent variable that indicates user intention. The intuition is that not all users who retweet a rumor have the malicious intention of spreading a harmful rumor. Sometimes a rumor may appear harmless and cannot be understood easily, and normal users may mistakenly spread them. With our approach, we can distinguish between malicious users and normal users even though they both retweeted rumors (or true stories). This distinction is particularly useful when the stories appear to be misleading and are difficult to judge. To summarize, our main contributions with this work include:

- We model user intention and story veracity in a same framework. This approach allows us to detect malicious users and predict rumor more accurately, particularly for stories whose veracity is difficult to judge. To the best of our knowledge, this is the first work that models user intention, user maliciousness and story veracity in the same framework.
- We develop algorithms to learn the user maliciousness and story veracity. More specifically, we design a Gibbs sampling algorithm for learning user attributes from labeled stories, and an Expectation Maximization algorithm for learning the veracity of unlabeled stories.
- We test our approach on real-world rumor datasets, comparing it with a number of baseline approaches. The evaluation results are consistent with our initial intuition, that our approach is more powerful for rumors that are confusing and difficult to judge. But it shows weakness for rumors that are easy to recognize.

2. Related Work

Our work follows two recent research trends regarding social media, namely, rumor detection and user attribute discovery. We first introduce the related literature about rumor detection. Castillo et al. investigated the topic of information credibility on Twitter, and proposed a prediction model based on a list of features [14]. They separated tweets into news and chats, and studied the credibility of news tweets. News credibility was automatically determined using supervised machine learning models such as SVM and decision trees, built upon message-based, user-based, topic-based, and propagation-based features. While the focus of this work was the perception of credibility, instead of the actual veracity of messages, it has set the standard for a line of researches, and its methodology has been repeatedly used in later works regarding message veracity on social media. For example, Kwon et al. investigated rumor detection on social media using a similar method [12]. Based on a dataset labeled as rumor and non-rumor, they tested the features proposed by Castillo et al. as well as three categories of new features, temporal, structural, and linguistic, together with machine learning models such as SVM and random forest. In a later work they also investigated feature stability over time with regard to rumor detection [15]. Yang et al. studied story veracity on a Chinese microblog platform, Sina Weibo, with new features including client-based and location-based features [9]. Comparing to Castillo features, their approach gained a substantial accuracy increase. Liu et al. extended this work by a new set of features called verification features, which included source credibility, source identification, source diversity, source and witness location, event propagation, and belief identification [13]. Zhang et al. investigated story veracity classification within the health domain, and reported correlation of features such as mention of numbers, the source of the rumor, and hyperlinks, with story veracity [16]. Ma et al. investigated rumor detection based on time series constructed from existing features, and showed improved accuracy [17]. As we can see, most of existing works used a similar method of constructing features and applying them on common machine learning models such as SVM and random forest. However, they did not provide insights of particular features, especially user-related attributes. We argue that our work that specifically models latent user attributes into the framework provides more interesting insights and may lead to better prediction accuracy.

We next introduce related literature regarding user profiling on Twitter. There is a number of works

that use supervised learning for discovering latent user attributes [18, 19, 20]. Typically, in these works, there is a training dataset labeled based on the attributes to be learned, such as age, regional origin, and political orientation. However, our work follows more closely the probabilistic approach for learning latent user attributes. For example, Barbera used a latent variable to represent users' political ideal point and proposed to learn the latent variable in a framework that considers follower and following connections [21]. Han and Tang proposed a variable indicating roles of a user in a community and learned the variable jointly with data and some other parameters [22]. The learning method was based on Gibbs sampling and EM, similar to our work. However, their work cannot be used to address the problem of discovering user intentions in message propagation. Although outside social media study, a work by Zhao et al. proposed an idea of two-sided perception of truth similar to our work [23]. Their framework was based on websites that show conflicting data to the same subject, which is common, for example, in movie review websites. They considered that the website credibilities for true positive and true negative were independent, and were treated separately in their learning framework. We cannot apply their work in our scenario, however, because the framework was not modeled after multiple users propagating the same story. Our model of two-sided perception of rumors, learned in retweeting data, is unique, based on our literature review.

3. Framework Overview and User Type Mapping

Our probabilistic model for malicious user and rumor detection is placed in a framework that detects rumors. This framework takes Twitter postings and clusters them into stories, each consists of a number of retweets. Some stories are manually assigned labels of either rumor or true story. The output of the framework is a set of prediction for unlabeled stories. The overview of the framework is shown in Figure 1. The core technique of learning user maliciousness and story veracity will be presented in the next section. In this section, we will briefly introduce the pre-processing step of user type mapping.

On Twitter, many users do not tweet often [1]. When we examine retweeting behavior, we find a user sparsity problem that many users only retweeted once in the dataset. It is very difficult to learn story veracity based on retweeting users who appear only once, and our solution is to generalize users into a discrete number of user types. Consequently, the model learns the story veracity not based on users, but on user types, which carry more information.

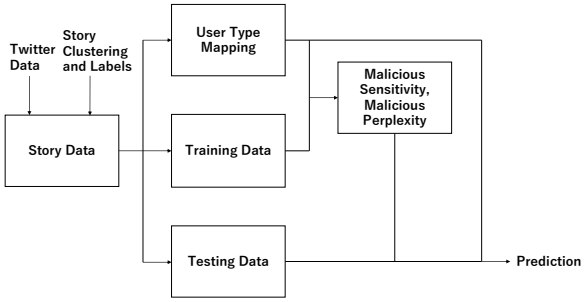


Figure 1. Rumor detection framework overview

Here we follow an unsupervised approach based on user attributes and clustering. We follow an existing work and extract six rumor-related attributes from user data [24]. These include presence of extreme keywords in the profile, following count, tweeting frequency, percentage of tweets containing topic-related keywords, percentage of tweets containing pictures, and percentage of tweets containing extreme keywords. The choice of extreme keywords and topic-related keywords should be based on the dataset. To demonstrate, the extreme keywords and topic-related keywords for the political rumor dataset, to be presented in Section 6, are shown in Table 1.

Table 1. Keywords for the experimental dataset

extreme keywords	tcot, ccot, tlot, ocra, pjnet, #p2, nobama, nohillary, stophillary, wakeupamerica, ohhillno, 4prison, #stop,
topic-related keywords	presiden, hillary, clinton, hart, obama, bush, rush, cruz, trump, conservat, democra, republic, socialis, senator, carson, rubio, fiorina, walker, kasich, huckabee, christie, paul, perry, santorum, chafee, malley, sanders, webb, biden, bloomberg, gore, poll, gilmore, pataki, jindal, everson, election, govern, pac, p.a.c, gop, g.o.p,

After transforming users into six-dimension feature vectors, we run k-means clustering to generate K user types. As a robust unsupervised clustering algorithm, k-means clustering has previously used in other user categorization tasks such as spammer detection [20]. We note that k-means is not the only solution for clustering users. Since our focus is not on user clustering but the probabilistic model, we choose this method due to its simplicity and efficiency, while other user clustering methods can also easily fit into the framework. We use

the R implementation of k-means[2].After clustering, each user is now mapped to a user type. For the reminder of this paper, we will not consider individual users, and will use *user* and *user type* interchangeably.

4. Reliability: A Simple Approach

Before we move into a more complex model, we would like to introduce a simple approach, which is based on a single user attribute, called *reliability*. The rationale behind this approach is that users who retweets rumors will likely be retweeting rumor again. And a story that is retweeted mostly by users who retweet rumor is likely to be a rumor. The reliability attribute describes how probable a user will retweet a rumor. Similar settings have previously been used for detecting faulty sensors in sensor network researches [25].

Algorithmically, let C_u^R be the number of times user u retweets a rumor, and C_u be the total number of retweets from user u. The reliability RLB_u is calculated as:

$$RLB_u = \frac{C_u^R}{C_u}.$$

Then we can derive a rumor score of a story based on the reliability of users who retweet it. Specifically, for a story which consists of N retweets, the rumor score is calculated as:

$$score = \frac{\sum_i RLB_u}{N}$$

where RLB_u is the reliability of user u who made retweet i. Finally, a threshold $thres$ is set so that if $score > thres$ then the story is predicted to be a rumor, and otherwise a true story.

This approach of using reliability to represent the likelihood a user would retweet a rumor is arguably reasonable and easy to implement. However, in practical situations, considering every retweet of a rumor as of malicious intend is over simplifying. In the next section we will introduce our probabilistic model that describes not only the behavior of retweeting rumors, but also the retweet intentions.

5. Malicious User and Rumor Detection

In this section, we present our probabilistic model of malicious user and rumor detection. We will first introduce the concept of separate perceptions for rumor and true stories, which supports the model. Then we will introduce a method for learning the model from labeled stories, followed by a method for veracity prediction for unlabeled stories.

5.1. Separate Perceptions for Rumors and True Stories

When considering user attributes related to rumor retweeting behavior, we intuitively think of maliciousness. We first consider an extreme case that a purely malicious user who would retweet and only retweet rumors. On the other hand, a normal user is a user who would never retweet a rumor. However, in reality, retweeting behavior is complex. Not all users retweeting a rumor are malicious users, and not all stories retweeted by a malicious user are rumors. We consider the following retweeting behaviors in our model.

- A malicious user sees a rumor, understands that it is a harmful rumor, and decides to retweet it.
- A normal user sees a rumor, but could not see that it is a harmful rumor, considers it as a true story instead, and retweets it.
- A malicious user sees a normal story, but due to confusion he considers it as a harmful rumor and retweets it.

We make an assumption by observing these behaviors that users have different perceptions for rumors and true stories. In addition to maliciousness, the perception of rumor and true story together decides whether a user will retweet a rumor or a true story. Therefore we propose two latent attributes. The first attribute is *malicious sensitivity*, denoted as θ, which describes the likelihood a user retweets a rumor as a rumor. The value $1 - \theta$ describes user ignorance that leads to retweeting rumor as true story. The second attribute is *malicious perplexity*, denoted as ϕ, which describes the likelihood a user retweets a true story as a rumor. The value $1 - \phi$ describes the likelihood of the normal behavior of retweeting a true story as a true story.

We then propose a latent variable that indicates the retweet intention. This latent variable, denoted as $\mathbf{z}$, is corresponding to the retweets of stories, and is influenced by θ and ϕ, as well as the story veracity. We set $z_{ij} = \{1, 0\}$ depending on if a retweet i of story j is intended as a rumor or a true story.

5.2. A Probabilistic Model for Malicious User and Rumor Detection

We now describe our probabilistic model based on two user attributes, namely, malicious sensitivity and

[2]https://stat.ethz.ch/R-manual/R-devel/library/stats/html/kmeans.html

malicious perplexity, denoted as θ and ϕ, respectively, and the latent variable $\mathbf{z}$, where $z_{ij} = 1$ means retweet i of story j is intended as a rumor, and $z_{ij} = 0$ means retweet i of story j is intended as a true story. We assume Beta distribution as the prior distribution for each θ_i, and ϕ_i for user i:

$$\theta \sim Beta(\beta_{0,0}, \beta_{0,1}),$$
$$\phi \sim Beta(\beta_{1,0}, \beta_{1,1}).$$

We also have a labeled dataset D with labels $l = \{r, t\}$, where a subset D^r contains only rumors, and a subset D^t contains only true stories. The dataset D consists of M stories, each in turn consists N retweeting users. The latent variable $\mathbf{z}$, corresponding to retweets $\mathbf{r}$, is thus depending on two user parameters and the label l. The dependency relationships of these parameters and variables are demonstrated in Figure 2, where the observed variables are shaded.

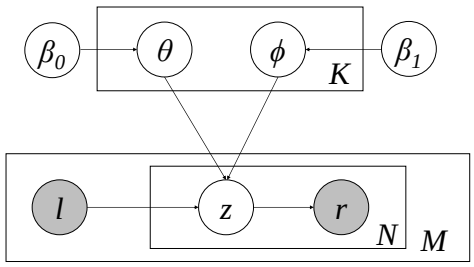


Figure 2. Graphical model of MURD

We generate retweet intention $\mathbf{z}$ as a Bernoulli distribution with θ and ϕ as the parameters. Specifically, when the story is a rumor ($l_j = 1$),

$$z_{ij} \sim Bernoulli(\theta_i)$$

and when the story is a true story ($l_j = 0$),

$$z_{ij} \sim Bernoulli(\phi_i).$$

5.3. Model Learning

The likelihood of the data is given as the following.

$$\begin{aligned} p(D|\mathbf{z}, \theta, \phi) = & \prod p(D^r|\mathbf{z}, \theta) \times \prod p(D^t|\mathbf{z}, \phi) \\ = & \prod_{i,j \in D^r} \left[\theta_i^{z_{ij}} \times (1-\theta_i)^{1-z_{ij}}\right] \\ & \times \prod_{i,j \in D^t} \left[\phi_i^{z_{ij}} \times (1-\phi_i)^{1-z_{ij}}\right]. \end{aligned} \quad (1)$$

Learning the model means finding the parameters that will give the maximum likelihood of the above formula. We will not attempt to solve the maximization problem analytically, since there are too many parameters and derivatives are intractable. Instead we will use a Markov Chain Monte Carlo (MCMC) method to approximate probability distribution of the parameters, which is a popular method for solving probabilistic models [26]. More specifically, we will use Gibbs sampling [27]. Gibbs sampling is a type of MCMC method that does not set a transit distribution in random walks, but instead randomly defines one parameter at a time assuming other parameters are given. It has been shown that Gibbs sampling can approximate parameter distributions faster and more accurate when the model contains a large number of parameters [23].

The main focus of our Gibbs sampling algorithm is the latent variable $\mathbf{z}$. Since θ and ϕ can be easily derived from $\mathbf{z}$, they can be omitted in the algorithm. Moreover, the probability of $\mathbf{z}$ can be derived from counts of retweets in different veracity and intention categories.

Let us use C_u^R to denote the count of retweets of rumors by user u, and C_u^T the count of retweets of true stories. Furthermore, CR_u^R denotes the count of retweets of rumors intended as rumors by user u, and CR_u^T the count of retweets of true stories intended as rumors, both of which are derived from $\mathbf{z}$. The posterior probability of z_{ij} from user u can then be derived as:

$$p(z_{ij} = 1|l_j = r, \theta) = \frac{CR_u^R + \beta_{0,0}}{C_u^R + \beta_{0,0} + \beta_{0,1}},$$
$$p(z_{ij} = 1|l_j = t, \phi) = \frac{CR_u^T + \beta_{1,0}}{C_u^T + \beta_{1,0} + \beta_{1,1}}.$$

Our implementation of the Gibbs sampling is shown in Algorithm 1. The main function of the algorithm is to assign z_{ij} one by one assuming all other $\mathbf{z}$ that are not z_{ij} are given (line 4-20). The random walk is achieved by setting z_{ij} randomly using the calculated probability (7-17). As a typical MCMC algorithm setting, at fix intervals after burn-in iterations, θ and ϕ are updated (line 21-24). The final values for θ and ϕ are the average values of multiple updates.

Finally, the maliciousness user attribute can be derived from $\mathbf{z}$. We can simply use the ratio of counts of retweets intended as rumors in all retweets:

$$maliciousness_u = \frac{CR_u^R + CR_u^T}{C_u^R + C_u^T}.$$

Algorithm 1 Gibbs sampling for learning users maliciousness

1: $s \leftarrow$ sample size
2: randomly assign {0, 1} to $\mathbf{z}$
3: **for** $iter$ in n learning iterations **do**
4: **for** each story j with label l **do**
5: **for** each retweet i from user u **do**
6: update counters removing z_{ij}
7: **if** $l = r$ **then**
8: $p \leftarrow \frac{CR_u^R + \beta_{0,0}}{C_u^R + \beta_{0,0} + \beta_{0,1}}$
9: **else**
10: $p \leftarrow \frac{CR_u^T + \beta_{1,0}}{C_u^T + \beta_{1,0} + \beta_{1,1}}$
11: **end if**
12: $c \leftarrow$ draw from Uniform[0, 1]
13: **if** $c < p$ **then**
14: $z_{ij} \leftarrow 1$
15: **else**
16: $z_{ij} \leftarrow 0$
17: **end if**
18: update counters adding z_{ij}
19: **end for**
20: **end for**
21: **if** $iter > burnin$ and $iter\%thin = 0$ **then**
22: $\theta \leftarrow \theta + \frac{1}{s} \times \frac{CR_u^R + \beta_{0,0}}{C_u^R + \beta_{0,0} + \beta_{0,1}}$
23: $\phi \leftarrow \phi + \frac{1}{s} \times \frac{CR_u^T + \beta_{1,0}}{C_u^T + \beta_{1,0} + \beta_{1,1}}$
24: **end if**
25: **end for**

5.4. Inferring Unlabeled Stories

We have used the Gibbs sampling for learning user attributes θ and ϕ given labeled stories. However, we have not shown how to predict veracity of unlabeled stories. In this section we will present a story veracity prediction method based on the user attributes learned from the training data. Our method is based on Expectation Maximization (EM), which is a popular algorithm for optimizing models involving latent variables [28, 29]. A EM algorithm typically involves alternating between two steps. In the *Expectation* step, the most likely latent variable is set given model parameters. In the *Maximization* step, the parameters that maximize the model likelihood are chosen, given fixed latent variable. The algorithm will come to stop when the prediction converges.

In this part of veracity inference problem, we use the same latent variable $\mathbf{z}$ to indicate retweet intention. For model parameters, instead of user attributes, we now use l to represent the latent story veracity. User attributes θ and ϕ are taken as learned with the labeled training data.

Our EM algorithm is shown in Algorithm 2. First the

Algorithm 2 EM for learning new labels

1: initialize $pred$
2: **while** prediction not converged **do**
/ Expectation */*
3: **for** each story j with prediction $pred_j$ **do**
4: **for** each retweet i from user u **do**
5: **if** $pred_j = t$ **then**
6: $p \leftarrow \theta_u$
7: **else**
8: $p \leftarrow \phi_u$
9: **end if**
10: **if** $p > 0.5$ **then**
11: $z_{ij} \leftarrow 1$
12: **else**
13: $z_{ij} \leftarrow 0$
14: **end if**
15: **end for**
16: **end for**
/ Maximization */*
17: **for** each story j **do**
18: $s_r \leftarrow \log p(l = r|\mathbf{z})$ from Equation (2)
19: $s_t \leftarrow \log p(l = t|\mathbf{z})$ from Equation (3)
20: **if** $s_r > s_t$ **then**
21: $pred'_j \leftarrow 1$
22: **else**
23: $pred'_j \leftarrow 0$
24: **end if**
25: **end for**
26: compare $pred$ with $pred'$ and replace if necessary
27: **end while**

prediction is initialized, usually with all positives (line 1). In the expectation step, we assign values to $\mathbf{z}$ given the prediction and user attributes, based on the following formula:

$$p(z_{ij} = 1|pred_j = r, \theta_u) = \theta_u,$$
$$p(z_{ij} = 1|pred_j = t, \phi_u) = \phi_u$$

and if $p(z_{ij} = 1) > 0.5$ we set z_{ij} as 1, otherwise we set z_{ij} as 0 (line 5-14).

In the maximization step, the goal is to choose the parameter τ so that the likelihood $L(l; \mathbf{z})$ is maximized. To choose the parameter, we consider the probability formula:

$$p(l = r|\mathbf{z}) \propto p(\mathbf{z}|l = r)p(l = r)$$
$$\propto p(l = r) \prod_{ij} p(z_{ij}|l_j = r)$$

where

$$p(z_{ij}|l_j = r) = z_{ij} \cdot \theta_u + (1 - z_{ij}) \cdot (1 - \theta_u).$$

Similarly we have:

$$p(z_{ij}|l_j = t) = z_{ij} \cdot \phi_u + (1 - z_{ij}) \cdot (1 - \phi_u).$$

Since our goal is not to inference the exact probability but to determine which parameter provides higher probability, we can compare the log-likelihood, which is more computationally efficient. More specifically, we compare $\log p(l = r|\mathbf{z})$ and $\log p(l = t|\mathbf{z})$, where

$$\log p(l = r|\mathbf{z}) = \log p(l = r) + \sum_i \log[z_{ij} \cdot \theta_u + (1 - z_{ij}) \cdot (1 - \theta_u)], \quad (2)$$

$$\log p(l = t|\mathbf{z}) = \log(1 - p(l = r)) + \sum_i \log[z_{ij} \cdot \phi_u + (1 - z_{ij}) \cdot (1 - \phi_u)]. \quad (3)$$

After comparing the log-likelihood, we make a prediction depending on which label provides higher likelihood (line 18-24). Here the prior probability of positive label $p(l = r)$ needs to be set manually, although normally it would not have a strong impact on the prediction.

6. Experimental Analysis

In this section, we present our experimental analysis based on two real-world datasets. We will first explain the datasets, followed by evaluation metrics and compared baselines. Then we will discuss the experimental results.

6.1. Datasets

We use two previously studied rumor datasets that contain user information and retweeting structure as our experimental datasets [24]. Both datasets are collected just before the 2016 U.S. presidency election, and contain tweets regarding two political actors, Hillary Clinton and Barack Obama. The first dataset, called *news* dataset, contains a number of news stories that include a link to a news article. The second dataset, called *picture* dataset, contains a number of picture stories that include a link to an online picture, which usually shows texts describing a story. Both dataset contain veracity labels for a number of stories, manually assigned by examining the stories in external sources. Examples of news stories are shown in Table 2 while examples of picture stories are shown in Figure 3.

Table 2. Examples of labeled news stories

rumor

Hillary's email firm was run from an apartment with servers in the BATHROOM
The IT firm hired by Clinton to oversee her private server was reportedly 'a mom and pop shop'
OBAMA SPENT $770 MILLION TAXPAYER $$$ REBUILD & RENOVATE MOSQUES
Revelation of God: Obama took on through a human clone comes from Pharaoh Aka Obamunaki!

true story

@realDonaldTrump takes aim at Hillary Clinton in new video just released on Instagram
Grassley questions whether Clinton attorney had clearance for thumb drives
Tell the Obama Administration: Stop Garnishing Social Security to Pay Student Loan Debt. Sign the petition here
President Obama's New 'College Scorecard' Is A Torrent Of Data

The statistics of two datasets are shown in Table 3. In a total there are 390 labeled stories. From the statistics we can see that for news stories, the number of retweets is larger but the number of users is smaller, indicating repeating appearance of same users in different stories. In contrast, picture stories contain less retweets but more users.

Table 3. Statistics of datasets

	#stories	#rumors	#tweets	#users
news	132	35	18,517	3,640
picture	258	97	11,064	8,398

6.2. Evaluation Metrics and Compared Baselines

The purpose of this evaluation is to test the effectiveness of our method in predicting rumors. Thus we use precision, recall, and f1 measures against the rumor prediction. The precision is calculated as $\frac{TP}{TP+FP}$ and recall $\frac{TP}{TP+FN}$, where TP, FP, and FN are the true positives, false positives and false negatives in the prediction with regard to rumor labels. As the overall accuracy indicator, f1 is calculated as $2 \times \frac{precision \times recall}{precision + recall}$. We run 5-fold cross validation and take the average result of five runs for each compared method.

We compare the performance of the propose MURD model with four baseline methods, namely, Castillo, Liu, Bag-of-User (BOU), and RLB. Castillo et al. has tested a list of features with regard to their effectiveness in predicting message credibility, and has been used as a popular baseline method in later works regarding message veracity [14]. From the list of features, the top-4 features are revealed to be user features, including

(a) rumors

(b) true stories

Figure 3. Examples of labeled picture stories

average registered age, average status count, average number of followers, and average number of friends. For comparison, we extract these four user features from our data. For each story retweeted by N users, the feature is the average feature value of N users. We use random forest as the learning model, which has been shown to have a good accuracy when used with Castillo features.

As an extension to Castillo features, Liu et al. propose an additional feature set, which is shown to have improved the prediction accuracy [13]. As the second baseline, we use four user features from Liu feature set, including ratio of profiles that contains top domain URL, person names, location names, and professions, respectively. We handle this feature set in the same way as the first baseline, and use random forest as the learning model.

Given that a story includes a number of retweeting users, a way to view this data is to consider it as a Bag-of-Users (BOU), similar to Bag-of-Words in text document analysis. For this baseline, we generate a feature vector for each story, based on the count of users who retweeted the story. The dimension of the vector is the same as the number of classes used in the k-means clustering of users. For the learning model we use logistic regression, since we find that it has superior performance comparing to SVM and random forest.

The forth baseline is the reliability-based approach (RLB) we introduced in section 4. We use the median score as the decision threshold, which means half of the test data will be predicted as rumors and the other half as true stories. Without prior knowledge of the ratio of rumors in the dataset, we consider 0.5 as a reasonable estimation.

6.3. Rumor Detection Results

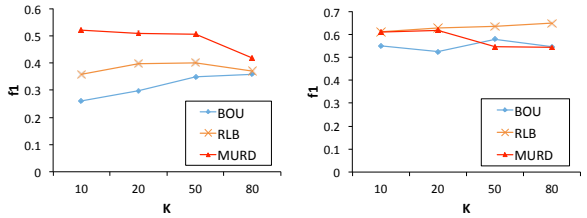
The prediction accuracy results of the proposed and baseline methods are shown in Table 4. The best score for each metric is highlighted in bold font. Results of the proposed MURD model are shown in the last column. The last row shows the overall accuracy by averaging f1 scores for the two datasets. As we can see from the results, the proposed MURD model is significantly better for the news dataset. For the picture dataset, Liu features works the best. However, its performance is poor for the news dataset. On average, MURD is shown to be superior compared to the baseline methods.

Table 4. Accuracy results of proposed and baseline methods

	Castillo	Liu	BOU	RLB	MURD
news					
prec	0.399	0.459	**0.485**	0.300	0.437
recall	0.185	0.177	0.216	0.580	**0.610**
f1	0.253	0.255	0.299	0.396	**0.509**
picture					
prec	0.706	**0.759**	0.543	0.556	0.529
recall	0.692	0.706	0.506	0.719	**0.743**
f1	0.699	**0.732**	0.524	0.627	0.618
average					
f1	0.476	0.493	0.411	0.511	**0.563**

We also investigate the method performance with regard to different number of classes in the user mapping. The results are shown in Figure 4. Among tested methods, MURD is shown to be steadily better for the news dataset regardless of K values. However, for the picture dataset it shows mixed performance, and surprisingly the RLB method, given its simplicity, performs best. The effect on prediction accuracy by changing K is not obvious, although we can tell the performance is generally optimal when $K = 20$.

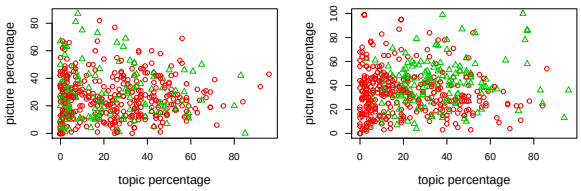
Here we give an explanation of why MURD achieves good results for the news dataset but not so good for the picture dataset. Figure 5 shows clustering of news and picture retweets. We randomly plot 500 retweets from both datasets, in which the two axis are two user attributes, and two different symbols indicating if the story retweeted is a rumor or a true story. We can see clearly that the separation between users who retweeted



(a) news (b) picture

Figure 4. Accuracy results with different K values

rumors and users who retweeted true stories is much more visible for the picture dataset than for the news dataset. What it implies is that users retweeting rumors and true stories are more similar for news stories than for picture stories, and the reason is that rumors in the news dataset are more difficult to be recognized than in the picture dataset, and both malicious users and normal users retweet news rumors. Consequently, consistent with our intuition, when the stories are complex and confusing, as in the news dataset, MURD can be much more powerful by learning the intention latent variable, which cannot be captured by the baseline methods.



(a) news (b) picture

Figure 5. Clustering of news and picture retweets

6.4. Malicious User Analysis

In this section, we provide an analysis of the effectiveness of our approach in capturing malicious users. This analysis is based on 20 user types and the news stories dataset. We compare user maliciousness learned by MURD with the coefficient learned by the logistic regression on user types (the BOU method), which shows which user type correlate most positively to the rumor. The results for the two approaches are quite different. We remove some user types containing too few users (< 30). In the remaining 9 user types ranked by maliciousness, the most malicious user type ranked by MURD is the 5th ranked by BOU, and the most malicious user type ranked by BOU is the 4th ranked by MURD.

We compare the users who are judged as the most malicious by MURD and BOU methods. We take the user profile text of the users in the type found by both methods and count the frequencies of words within the text. We give each word a significance score as $\frac{\text{frequency within the type}}{\text{frequency in all user profiles}}$, where the higher the score, the more strongly the word is associated with the type. The top words of the most malicious user type found by MURD and BOU are shown in Table 5. As we can see, the profile of the most malicious user type found by MURD contains many aggressive words that reveal the malicious characteristics of the user, while that discovered by BOU appears to be rather neutral. This result shows that MURD can detect malicious users more effectively.

Table 5. Keywords associated with detected malicious user types

MURD	#tgdn, #pjnet, #ccot, military, #tcot, constitutional, news, tweets, #2a, israel
BOU	fans, media, account, twitter, com, http, follow, father, christian, conservative

7. Conclusion

In this paper we propose a probabilistic model for malicious user and rumor detection (MURD). In contrast to existing approaches, we model not only the behavior but also the intention when a user retweets a rumor or a true story. This approach helps us capture more accurately user maliciousness and rumor veracity, especially when the stories are complex and confusing. In experiments we use two datasets. The news dataset contains stories that are more confusing, while the picture dataset contains stories that are easier to understand. The evaluation results show that MURD achieves superior rumor detection accuracy for the news dataset, which is consistent with our intuition. We also show with examples that MURD can be quite effective in capturing malicious users. In this work, we have purposefully left out the content of the story. Further implementation can additionally examine story contents using existing techniques to achieve better detection accuracy with combined results. Another future direction maybe investigating retweet of stories across different social media platforms.

Acknowledgement

This research is partially supported by JST CREST Grant Number J181401085.

References

[1] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts, "Who says what to whom on Twitter," in *Proceedings of the 20th International World Wide Web Conference*, pp. 705–714, 2011.

[2] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, "Predicting elections with Twitter: What 140 characters reveal about political sentiment," in *Proceedings of the Fourth International Conference on Weblogs and Social Media*, pp. 178–185, 2010.

[3] K. Lerman and R. Ghosh, "Information contagion: An empirical study of the spread of news on digg and Twitter social networks.," *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pp. 90–97, 2010.

[4] E. T. K. Sang and J. Bos, "Predicting the 2011 dutch senate election results with Twitter," in *Proceedings of the Workshop on Semantic Analysis in Social Media*, pp. 53–60, 2012.

[5] J. Maddock, K. Starbird, H. Al-Hassani, D. E. Sandoval, M. Orand, and R. M. Mason, "Characterizing online rumoring behavior using multi-dimensional signatures," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pp. 228–241, 2015.

[6] V. Qazvinian, 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, 2011.

[7] C. Budak, D. Agrawal, and A. El Abbadi, "Limiting the spread of misinformation in social networks," in *Proceedings of the 20th international conference on World Wide Web*, pp. 665–674, ACM, 2011.

[8] K. Starbird, J. Maddock, M. Orand, P. Achterman, and R. M. Mason, "Rumors, false flags, and digital vigilantes: Misinformation on Twitter after the 2013 boston marathon bombing," in *iConference 2014 Proceedings*, pp. 654–662, iSchools, 2014.

[9] F. Yang, Y. Liu, X. Yu, and M. Yang, "Automatic detection of rumor on sina weibo," in *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, p. 13, ACM, 2012.

[10] S. Kiousis, "Public trust or mistrust? perceptions of media credibility in the information age," *Mass Communication & Society*, vol. 4, no. 4, pp. 381–403, 2001.

[11] M. Kakol, M. Jankowski-Lorek, K. Abramczuk, A. Wierzbicki, and M. Catasta, "On the subjectivity and bias of web content credibility evaluations," in *Proceedings of the 22nd international conference on World Wide Web companion*, pp. 1131–1136, 2013.

[12] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent features of rumor propagation in online social media," in *Proceedings of 13th International Conference on Data Mining*, pp. 1103–1108, 2013.

[13] X. Liu, A. Nourbakhsh, Q. Li, R. Fang, and S. Shah, "Real-time rumor debunking on twitter," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 1867–1870, ACM, 2015.

[14] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on Twitter," in *Proceedings of the 20th International World Wide Web Conference*, pp. 675–684, 2011.

[15] S. Kwon, M. Cha, and K. Jung, "Rumor detection over varying time windows," *PloS one*, vol. 12, no. 1, p. e0168344, 2017.

[16] Z. Zhang, Z. Zhang, and H. Li, "Predictors of the authenticity of internet health rumours," *Health Information & Libraries Journal*, vol. 32, no. 3, pp. 195–205, 2015.

[17] J. Ma, W. Gao, Z. Wei, Y. Lu, and K.-F. Wong, "Detect rumors using time series of social context information on microblogging websites," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 1751–1754, ACM, 2015.

[18] D. Rao, D. Yarowsky, A. Shreevats, and M. Gupta, "Classifying latent user attributes in Twitter," in *Proceedings of the 2nd International Workshop on Search and Mining User-generated Contents*, pp. 37–44, 2010.

[19] M. Pennacchiotti and A.-M. Popescu, "A machine learning approach to twitter user classification," in *Proceedings of the Fifth International Conference on Weblogs and Social Media*, pp. 281–288, 2011.

[20] Z. Miller, B. Dickinson, W. Deitrick, W. Hu, and A. H. Wang, "Twitter spammer detection using data stream clustering," *Information Sciences*, vol. 260, pp. 64–73, 2014.

[21] P. Barberá, "Birds of the same feather tweet together: Bayesian ideal point estimation using Twitter data," *Political Analysis*, vol. 23, no. 1, pp. 76–91, 2015.

[22] Y. Han and J. Tang, "Probabilistic community and role model for social networks," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 407–416, ACM, 2015.

[23] B. Zhao, B. I. Rubinstein, J. Gemmell, and J. Han, "A Bayesian approach to discovering truth from conflicting sources for data integration," *Proceedings of the VLDB Endowment*, vol. 5, no. 6, pp. 550–561, 2012.

[24] C. Chang, Y. Zhang, C. Szabo, and Q. Z. Sheng, "Extreme user and political rumor detection on twitter," in *Proceedings of the 12th International Conference on Advanced Data Mining and Applications*, pp. 751–763, Springer, 2016.

[25] Y. Zhang, C. Szabo, and Q. Z. Sheng, "Cleaning environmental sensing data streams based on individual sensor reliability," in *Proceedings of the 15th International Conference on Web Information Systems Engineering, Part II*, pp. 405–414, 2014.

[26] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, "An introduction to mcmc for machine learning," *Machine learning*, vol. 50, no. 1-2, pp. 5–43, 2003.

[27] S. L. Zeger and M. R. Karim, "Generalized linear models with random effects; a gibbs sampling approach," *Journal of the American Statistical Association*, vol. 86, no. 413, pp. 79–86, 1991.

[28] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[29] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, "On truth discovery in social sensing: A maximum likelihood estimation approach," in *Proceedings of the 11th International Conference on Information Processing in Sensor Networks*, pp. 233–244, 2012.