Detecting Marionette Microblog Users for Improved Information Credibility

Xian Wu[1], Ziming Feng[1] Wei Fan[2], Jing Gao[3], and Yong Yu[1]

[1] Shanghai Jiao Tong University, Shanghai, 200240, P.R.China
{wuxian,fengzm,yyu}@apex.sjtu.edu.cn
[2] Huawei Noah's Ark Lab, Hong Kong
david.fanwei@huawei.com
[3] University at Buffalo, NY 14260, USA
jing@buffalo.edu

Abstract. In this paper, we mine a special group of microblog users: the "marionette" users, who are created or employed by backstage "puppeteers", either through programs or manually. Unlike normal users that access microblogs for information sharing or social communication, the marionette users perform specific tasks to earn financial profits. For example, they follow certain users to increase their "statistical popularity", or retweet some tweets to amplify their "statistical impact". The fabricated follower or retweet counts not only mislead normal users to wrong information, but also seriously impair microblog-based applications, such as popular tweets selection and expert finding. In this paper, we study the important problem of detecting marionette users on microblog platforms. This problem is challenging because puppeteers are employing complicated strategies to generate marionette users that present similar behaviors as normal ones. To tackle this challenge, we propose to take into account two types of discriminative information: (1) individual user tweeting behaviors and (2) the social interactions among users. By integrating both information into a semi-supervised probabilistic model, we can effectively distinguish marionette users from normal ones. By applying the proposed model to one of the most popular microblog platform (Sina Weibo) in China, we find that the model can detect marionette users with f-measure close to 0.9. In addition, we propose an application to measure the credibility of retweet counts.

Keywords: marionette microblog user, information credibility, fake followers and retweets

1 Introduction

The flourish of Microblog services, such as Twitter, Sina Weibo and Tencent Weibo, has attracted enormous number of web users. According to recent statistics, the number of Twitter users has exceeded 500 million in July 2012,[4] and

[4] http://semiocast.com/publications

Sina Weibo has more than 300 million users in Match 2012.[5] Such a large volume of participants has made microblog a new social phenomenon that attracts attention from a variety of domains, such as business intelligence, social science and life science.

In Microblog services, the messages (tweets) usually deliver time sensitive information, e.g., "What the user is doing now?". By following others, a user will be notified of all their posted tweets and thus keep track of what these people are doing or thinking about. Therefore, the number of followers measures someone's popularity, and can indicate how much influence someone has. For celebrities, a large number of followers shows their social impact and can increase their power in advertisement contract negotiations. As for normal users, a relatively large number of followers represents rich social connections and promotes one's position in social networks. Therefore, both celebrities and normal users are eager to get more followers.

Due to the retweet mechanism, information propagation is quite efficient in microblog services. Once a user posts a message, his followers will be notified immediately. If these followers further retweet this message, their followers can view it immediately as well. In this way, the number of audiences can grow at an exponential rate. Therefore, the retweet count of a message represents its popularity, as more users wish to share with their followers. On many microblog platforms (e.g., Sina Weibo), the retweet count is adopted as the key metric to select top stories.[6] As a result, some microblog users are willing to purchase more retweets to promote their messages for commercial purpose.

The desire for more followers and retweets triggers the emergence of a new microblog business: follower and retweet purchase. The backstage puppeteers maintain a large pool of marionette users. To purchase followers or retweets, the buyer first provides his user id or tweet id. Then the puppeteer activates certain number of marionette users to follow this buyer or retweet his message. The number of followers or retweets depends on the price paid. The fee is typically modest, 25 USD for 5,000 followers in Twitter, and 15 Yuan (i.e., 2.5 USD) for 10,000 followers in Sina Weibo. Moreover, the massive following process is quite efficient. For example, it only takes one night to add 10,000 followers in Sina Weibo, which can make someone become "famous" overnight.

From the perspectives of people who made the follower and retweet purchase, the marionette users can satisfy their needs to become famous and help in promoting commercial advertisements, but overall the fabrications conducted by marionette users can lead to serious damages:

- The purchased followers fabricate the social influence of users, and the purchased retweets amplify the public attention paid to the messages. As a result, the fake numbers can mislead real users and data mining applications based on microblog data, such as [1]
- Beside promoting advertisements, the marionette users are sometimes employed to distribute rumors [2]. It will not only mislead normal users but

[5] http://news.sina.com.cn/m/news/roll/2012-03-31/003724202903.shtml
[6] http://hot.weibo.com

also provide wrong evidence for business [3] and government's establishment of policies and strategies. Thus, this becomes a serious financial and political problem.
– To disguise as normal users, the marionette users are operated by puppeteers to perform some random actions, including following, retweeting and replying. Such actions can interfere operating and viewing experiences of normal users and result in unpleasant user experience.

Therefore, identifying marionette users is a key challenge for ensuring normal functioning of microblog services. However, marionette users are difficulty to detect. Monitoring the marionette users over the past two years, we find that the difficulty of detecting marionette users has gradually gone up. Back in November 2011, we purchased 2,000 followers from Taobao (China EBay) and all these fake followers were recognized by microblog services and deleted within two days. Such quick detection can be attributed to the several discriminative features. For example, the marionette accounts are usually created from the same IP address within a short period of time, many marionette users posted no original tweets but only performed massive following or retweeting, and so on. Therefore, the microblog services can employ simple rules to detect marionette users and delete their accounts. However, the marionette users are evolving and becoming more intelligent. Nowadays, the puppeteers hire people or use crowdsourcing to create marionette accounts manually. To make these accounts behave like normal users, the puppeteers develop highly sophisticated strategies that operate the marionette users to follow celebrities, reply to hot tweets, and conduct other complicated operations. These disguises can easily overcome the filtering strategies of microblog platforms and make marionette users much more difficult to be detected. In February 2012, we purchased another 4,000 followers. This time, 1,790 marionette users survived after five weeks, and around 1,000 marionette users are still active by Feb 2013.

After analyzing the behaviors of marionette users and comparing them with normal users, we find that the following two types of information are useful in detecting marionette users.

– Local Features: The features that describe individual user behaviors, which could be either textual or numerical. The local features can capture the different behaviors between normal and marionette users. For example, the following/follower counts are important features that distinguish a large portion of normal users from marionette users. The time interval between tweets and the tweet posting devices also serve as effective clues to detect marionette users.
– Social Relations: The following, retweeting or other relationships among users. Such relations provide important information for marionette user detection. For example, the marionette users will follow both normal users and other marionette users. They follow normal users to disguise or for profits, and follow other marionette users to help them to disguise. On the other hand, the normal users are less likely to follow marionette users. Therefore

the neighboring users that are connected by the "following" relation can be used to recognize marionette users.

The two types of features provide complementary predictive powers for the task of marionette user detection. Therefore, we propose a probabilistic model that seamlessly takes both the rich local features as well as social relations among users into consideration to detect marionette users more effectively. On the dataset collected from Sina Weibo, the proposed model is able to detect marionette users with the f-measure close to 0.9. As a result, we are able to measure the true popularity of hot tweets. For example, given a hot tweet, we can first extract the users who retweet it and then evaluate whether these users are marionette users. The percentage of normal users can be used to measure the true popularity.

2 The Proposed Model

In this section, we describe the proposed probabilistic model that integrates local features and social relations in marionette user detection.

2.1 Notation Description

Let u_i denote a microblog user and let the vector x_i denote the features of u_i. Each dimension of x_i represents a local feature, which could be the follower count of u_i or the tweeting device u_i has used before. Let the binary variable y_i denote the label of u_i, 1 stands for the marionette user and 0 stands for the normal user. Let $V^{(i)} = \{v_1^{(i)}, v_2^{(i)}, \ldots, v_{M(i)}^{(i)}\}$ denote the $M(i)$ users who are related to u_i. In microblog services, the social relations between users can be either explicit or implicit. To be concrete, "followed" and "following" are explicit relations while retweeting one's tweet or "mention" someone in a tweet establish implicit social relations. In this paper, we target to predict the label y_i of u_i given its local features x_i and his social relations $V^{(i)}$.

2.2 Problem Formulation

We will first describe how to only use local features that describe user behavior on the microblogging platform, such as follower/following counts, the posting devices, to build a discriminative model. We will later describe how to incorporate social relations into this model to further improve the performance. If we only consider the local features, marionette user detection is a typical classification problem. A variety of classification models can be used, among which we choose Logistic Regression because it can be adapted to incorporate social relations which will be shown later in this paper. We first describe how to model local user features using Logistic Regression model.

We introduce the sigmoid function in Eq.(1) to represent the probability of belonging to marionette or normal class given feature values, i.e., $P(y_i|x_i)$, for each user.

$$P_\theta(y_i|x_i) = h_\theta(x_i)^{y_i}(1 - h_\theta(x_i))^{(1-y_i)} \tag{1}$$

where $h_\theta(x_i) = \frac{1}{1+e^{-\theta^T x_i}}$ is equal to the probability that u_i is a marionette user. θ is the set of parameters that characterizes the sigmoid function. With Eq.(1), we can formulate the joint probability over N labeled users in Eq.(2), in which we try to find the parameter θ that maximizes this data likelihood.

$$\max_{\theta} \quad \prod_{i=1}^{N} P_\theta(y_i|x_i) \tag{2}$$

In the above formulation, each user is treated separately and the prediction of a marionette user only depends on one's local features. However, besides the local features, the relations between users are also discriminative for the task of predicting marionette users. To incorporate the social relations, we modify the objective function from Eq.(2) to Eq.(3).

$$\max_{\theta,\alpha} \quad \prod_{i=1}^{N} \{P_\theta(y_i|x_i) \prod_{j=1}^{M(i)} P_\alpha(y_i|y_j^{(i)})^d\} \tag{3}$$

In Eq.(3), we assume that, for each user u_i, the label of its $M(i)$ neighbors $y_0^{(i)}$, $y_1^{(i)}$, ..., $y_{M(i)}^{(i)}$ are known in advance. Then we can integrate the effect of local features and user connections together to predict marionette users. d is the coefficient that balances between the social relations and local features. The larger d is, the more biased the model is towards the social relations in making the predictions. Note that to simplify the presentation, we consider the case where only one type of social relations exists in Eq.(3). However, the proposed model is general enough and can be easily adapted to cover multi-type social relations. Take the microblog system for example, the common user relations include follower, following, mention, retweet and reply. We can introduce different parameter α to correspond to each kind of relation and model all relations in one unified framework.

In Eq.(3), $P_\theta(y_i|x_i)$ is formulated using the same sigmoid function shown in Eq.(1). $P_\alpha(y_i|y_j^{(i)})$ will be modeled using Bernoulli distribution and characterized by parameter α as shown in Eq.(4).

$$P_\alpha(y_i|y_j^{(i)} = k) = \alpha_k^{y_i}(1 - \alpha_k)^{(1-y_i)} \quad (k = 0, 1) \tag{4}$$

As k is either 1 or 0, we can write down all the possible $P_\alpha(y_i|y_j^{(i)} = k)$ in Eq.(5).

$$\begin{bmatrix} P(y_i=0|y_j^{(i)}=0)=\alpha_0 & P(y_i=1|y_j^{(i)}=0)=1-\alpha_0 \\ P(y_i=0|y_j^{(i)}=1)=\alpha_1 & P(y_i=1|y_j^{(i)}=1)=1-\alpha_1 \end{bmatrix} \tag{5}$$

For each user, the parameter α measures the influence received from his neighbors. α_0 indicates the chance of a user being a normal user if his neighbor is a normal user. If the neighbor is normal, the larger α_0 is, this user is more likely to be a normal user. Similarly, α_1 indicates the chance of a user being a normal user if his neighbor is a marionette user. If the neighbor is marionette, the larger α_1 is, this user is more likely to be a normal user. The logarithm of the joint probability in Eq.(3) can be represented in Eq.(6):

$$\begin{aligned}\ell(\theta,\alpha) = &\sum_{i=1}^{N} y_i \log h_\theta(x_i) + (1-y_i)\sum_{i=1}^{N}\log(1-h_\theta(x_i)) \\ &+d\sum_{i=1}^{N}\sum_{j=1}^{M(i)}\sum_{k=y_j^{(i)}}(y_i\log\alpha_k + (1-y_i)\log(1-\alpha_k))\end{aligned} \tag{6}$$

The model parameters θ and α will be inferred by maximizing the log-likelihood in Eq.(6). To solve this optimization problem, it is natural to apply gradient descent approaches. Notice that θ is only included in the first part of Eq.(6) and α is only included in the second part, we can maximize each part separately to infer θ and α. θ can be obtained via numerical optimization methods using the same procedure in the aforementioned Logistic Regression formulation. As for α, we can derive the following analytical solution by maximizing the following objective function.

$$\alpha_k = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M(i)}\sum_{k=y_j^{(i)}} y_i}{\sum_{i=1}^{N}\sum_{j=1}^{M(i)}\sum_{k=y_j^{(i)}} 1} \tag{7}$$

The above model takes social relations into consideration, but it has several disadvantages that may prevent its usage in real practice: First, the model only works in a supervised scenario where the class labels of all the neighbors of each user are observed. This is a strong assumption and can only be achieved by spending huge amounts of time and labeling costs to get sufficient training data. Second, even if we acquire sufficient labeled data, the discriminative information hidden in the labeled data is not fully utilized in the model. As shown in Eq.(3), the labels on a user's neighbors are only used in modeling $P(y_i|y_j^{(i)})$ without considering the relationship between the labels of these neighbors and their local features. Intuitively, if two neighbors have the same class label but different local features, their effect on the target user's label should be different.

Therefore, we propose to adapt Eq.(3) to Eq.(8) by considering both class labels and local features of a user's neighbors:

$$\max_{\theta,\alpha} \quad \prod_{i=1}^{N}\{P_\theta(y_i|x_i)\prod_{j=1}^{M(i)} P_{\alpha,\theta}(y_i|x_j^{(i)})^d\} \tag{8}$$

The only difference between Eq.(3) and Eq.(8) is that we replace $P_\alpha(y_i|y_j^{(i)})$ with $P_{\alpha,\theta}(y_i|x_j^{(i)})$. In this way, the proposed model incorporates the local features

of the neighbors and the model does not have the strong assumption that the neighbors' labels are fully observed.

In Eq.(8), we represent $P_\theta(y_i|x_i)$ using the same sigmoid function shown in Eq.(1). As for $P_{\alpha,\theta}(y_i|x_j^{(i)})$, its formulation can be inferred based on Eq.(9).

$$\begin{aligned} P_{\alpha,\theta}(y_i|x_j^{(i)}) &= \sum_{k=0}^{1} P_{\alpha,\theta}(y_i, y_j^{(i)} = k|x_j^{(i)}) \\ &= \sum_{k=0}^{1} P_{\alpha,\theta}(y_i|y_j^{(i)} = k, x_j^{(i)}) P_\theta(y_j^{(i)} = k|x_j^{(i)}) \end{aligned} \tag{9}$$

We assume that the label of a user y_i is conditionally independent of the local features $x_j^{(i)}$ of his neighbor given the label of this neighbor $y_j^{(i)}$, and thus we have $P_{\alpha,\theta}(y_i|y_j^{(i)} = k, x_j^{(i)}) = P_\alpha(y_i|y_j^{(i)} = k)$. Hence, we modify Eq.(9) accordingly into Eq.(10).

$$P_{\alpha,\theta}(y_i|x_j^{(i)}) = \sum_{k=0}^{1} P_\alpha(y_i|y_j^{(i)} = k) P_\theta(y_j^{(i)} = k|x_j^{(i)}) \tag{10}$$

By plugging the above definition of $P_{\alpha,\theta}(y_i|x_j^{(i)})$ into the proposed objective function in Eq.(8), we effectively integrate users and their neighbors' local features together with social relations in the discriminative model to distinguish marionette and normal users. Accordingly, the log-likelihood in Eq. (6) is modified to Eq. (11).

$$\begin{aligned} \ell(\theta, \alpha) = &\sum_{i=1}^{N} y_i \log h_\theta(x_i) + (1 - y_i) \sum_{i=1}^{N} \log(1 - h_\theta(x_i)) \\ &+ d \sum_{i=1}^{N} \sum_{j=1}^{M(i)} \log \sum_{k=0}^{1} P_\alpha(y_i|y_j^{(i)} = k) P_\theta(y_j^{(i)} = k|x_j^{(i)}) \end{aligned} \tag{11}$$

2.3 Parameter Estimation

In the proposed model, two sets of parameters need to be estimated: θ in both $P_\theta(y_i|x_j)$ and $P_{\alpha,\theta}(y_i|x_j^{(i)})$, and α in $P_{\alpha,\theta}(y_i|x_j^{(i)})$. These parameters should be obtained by maximizing the logarithm of Eq.(11). As the class labels of one's neighbors are unknown, we treat them as latent hidden variables during the inference procedure. The following hidden variable $z_{jk}^{(i)}$ is introduced in Eq. (12).

$$\begin{aligned} z_{jk}^{(i)} &\propto P_{\alpha,\theta}(y_i, y_j^{(i)} = k|x_j^{(i)})) \\ &\propto P_\alpha(y_i|y_j^{(i)} = k) P_\theta(y_j^{(i)} = k|x_j^{(i)}) \end{aligned} \tag{12}$$

Based on this hidden variable, the objective function in Eq.(11) can be represented in Eq.(13):

$$\ell'(z_{jk}^{(i)}, \theta, \alpha) = \sum_{i=1}^{N} \log P_\theta(y_i|x_i) + d \sum_{i=1}^{N} \sum_{j=1}^{M(i)} \sum_{k=0}^{1} z_{jk}^{(i)} \log P_\alpha(y_i|y_j^{(i)} = k) + d \sum_{i=1}^{N} \sum_{j=1}^{M(i)} \sum_{k=0}^{1} z_{jk}^{(i)} \log P_\theta(y_j^{(i)} = k|x_j^{(i)}) \quad (13)$$

We propose to use EM method to iteratively update model parameters and hidden variables. At the E-Step, the hidden variable $z_{jk}^{(i)}$ can be calculated via Eq.(14):

$$z_{jk}^{(i)} = \frac{P_\alpha(y_i|y_j^{(i)} = k) P_\theta(y_j^{(i)} = k|x_j^{(i)})}{\sum_{k=0}^{1} P_\alpha(y_i|y_j^{(i)} = k) P_\theta(y_j^{(i)} = k|x_j^{(i)})} \quad (14)$$

At the M-Step, we maximize the parameter $\ell'(z_{jk}^{(i)}, \theta, \alpha)$ with respect to α and get the following solution of α in Eq.(15).

$$\alpha_k = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M(i)} z_{jk}^{(i)} y_i}{\sum_{i=1}^{N} \sum_{j=1}^{M(i)} z_{jk}^{(i)}} \quad (15)$$

The estimation of θ can be transformed into the parameter estimation process of Logistic Regression by constructing a training set. Initially, the training data set only includes N labeled users $\{(x_1, y_1), \ldots, (x_N, y_N)\}$. Then for each neighbor of the users, two instances $(x_j^{(i)}, y_j^{(i)} = 0)$ and $(x_j^{(i)}, y_j^{(i)} = 1)$ are generated and added into the training data set. In total, there are $2 \sum_{i=1}^{N} M(i)$ new instances added. The weights of the newly added instances are different from those of the initial ones. For the initial training instance (x_i, y_i), its weight is 1, while the weight of the newly added instance $(x_j^{(i)}, y_j^{(i)} = k)$ is $d \times z_{jk}^{(i)}$. The detailed parameter estimation process is summarized in Algorithm 1.

After obtaining the values of α and θ using Algorithm 1 from data, we can now use the proposed model to predict the class label of a new user u_i. This user's label y_i can be predicted according to Eq.(16).

$$\arg\max_{y_i} \quad P_\theta(y_i|x_i) \prod_{j=1}^{M(i)} P_{\alpha,\theta}(y_i|x_j^{(i)})^d \quad (16)$$

where $P_\theta(y_i|x_i)$ can be calculated using Eq.(1) and $P_{\alpha,\theta}(y_i|x_j^{(i)})$ can be calculated using Eq.(10).

2.4 Time Complexity

Another perspective we want to discuss is the time complexity and the number of iterations needed to converge. As shown in Algorithm 1, the parameter estimation process basically consists of EM iterations. During each iteration, the

Algorithm 1: Parameter Estimation Process

Data: Training data set $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ and their unlabeled neighbors.

Result: Value of θ and α

while *EM not converged* **do**
  E step:
    Update $z_{jk}^{(i)}$ according to Eq.(14).
  M step:
    Update α according to Eq.(15).
    For each neighbor of each instance in D, add two instances $\{(x_j^{(i)}, y_j^{(i)} = k), k = 0, 1\}$ and assign weights $d \times z_{jk}^{(i)}$.
    Apply parameter estimation of Logistic Regression to calculate θ.

value of the hidden variable $z_{jk}^{(i)}$, θ and α are updated. According to Eq.(14) and Eq.(15), the time complexity for calculating $z_{jk}^{(i)}$ and α is $O(NM)$ where N is the number of instances and M is the average of the number of neighbors. As for θ, the calculation is the same as parameter estimation process of Logistic Regression, whose time complexity depends on the optimization method adopted. In total, the time complexity for training is $O(TNM + TL)$ where T denotes the number iterations and L represents the time complexity of Logistic Regression optimization.

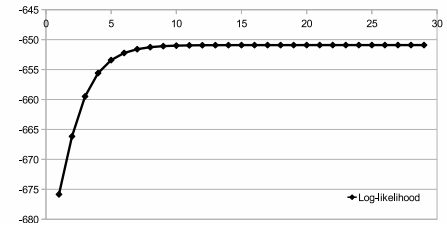


Fig. 1. The Log-likelihood Value with EM Iterations

We illustrate the convergence speed of the algorithm on the Weibo data set in Figure 1. We calculate the log-likelihood after each round of iteration and plot the values of log-likelihood with respect to each iteration. It can be observed that Algorithm 1 converges quickly. After 8 rounds, the log-likelihood becomes stable. Therefore, a small iteration number can achieve good performance. On the training data set consisting of 12,000 users with 30 iterations, the proposed approach only takes less than 10 seconds to converge on a commodity PC.

3 Experiments

In this section, we evaluate the proposed probabilistic model from two perspectives: (1) we calculate the classification accuracy and show that incorporating social relations can indeed improve the performance; (2) we demonstrate an application which measures the credibility of hot tweets with the suspicion of marionette user promotion.

3.1 Data Sets

Classification Corpus We acquire a data set that consists of labeled marionette and normal users to evaluate the proposed model.

- **Marionette Users**: To collect the corpus of marionette users, we first created three phishing Sina Weibo accounts and bought followers from three Taobao shops for three times. Each time we purchased 2,000 followers and altogether there are 6,000 in total. The first purchase was made on November 2011 and the other two were made on February 2012. On Feb 2013, one year after the purchase, we re-examined these bought marionette users and found that around 1,000 are still active while the rest have already been deleted or blocked by Sina Weibo. Over 1/6 marionette users are not discovered by Sina weibo for over a year. To target a more challenging problem and compensate the existing detecting methods of Sina Weibo, we select these well hidden marionette users into our corpus.
- **Normal Users**: As for the normal users, we first select several seed users manually and crawl the users that they are following. After that, the crawled users are taken as new seeds to continue the crawling. Through this iterative procedure, we collect users whose identifications have been verified by Sina Weibo. As Sina requires the users to fax their ID copies for verification, we are confident these users are normal users. From these verified users, we randomly select 1,000 into our corpus that is the same amount as the marionette users. In real life, the distribution of normal and marionette users is usually imbalanced. However, to make the classifier more accurate, we decide to under sample the normal users and use a balanced training set to train the classifier which is commonly used in imbalanced classification [4].

For each obtained user, we further randomly select 5 users from all their followers into the data set. As a result, this data set consists of 2,000 labeled users and 10,000 unlabeled users. The profiles and posted tweets of all these 12,000 users are crawled.

Suspicious Hot Tweet Corpus In Sina Weibo, the account named "social network analysis"[7] listed several hot tweets that were suspiciously promoted by marionette users. This account visualized the retweeting propagations of these

[7] http://weibo.com/dmonsns

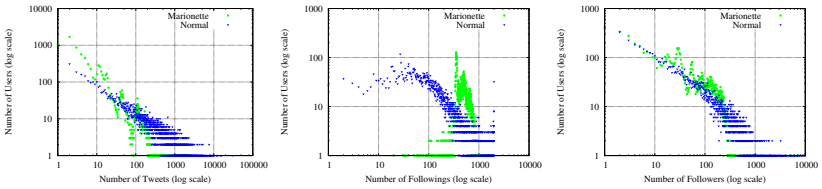
suspicious tweets and identified the topological differences compared with the normal hot tweets. For each suspicious tweet mentioned by this account, we first retrieve the list of users who have retweeted this tweet and randomly select 200 users to crawl their profiles, posted tweets as well as that of their 5 neighbors.

3.2 Feature Description

In this subsection, we analyze some local features of microblog users whether they are discriminative in marionette user classification.

Number of Tweets/Followings/Followers For each user, we extract the number of their posted tweets, the number of their followings and followers, and demonstrate the comparison results in three sub-figures of Figure 2 respectively. The x-axis represents different numbers of tweets, followings and followers, while the y-axis represents the number of users with the same number of tweets, followings and followers. Both axis are in the logarithmic scale.

In Figure 2(a), we find the marionettes are relatively inactive in tweeting, a large proportion marionettes post less than 20 tweets. On the contrary, the normal users are more active. The most "energetic" normal user posts more than 30,000 tweets. Therefore, a large number of tweets can be an effective feature to recognize normal users; In Figure 2(b), we find the number of followings of most marionettes lies between 100 to 1,000. One possible explanation to this range is that the puppeteer restricts the maximal following times to avoid being detected by microblog services.



(a): Tweet (b): Followings (c): Followers

Fig. 2. User Number Distribution on Number of Tweets, Followings and Followers

Tweet Posting Device Microblog services provide multiple access manners, including web interface, mobile clients, third party microblog applications, etc. Thus, we try to figure out whether there are any differences in posting devices between normals and marionettes. All the tweets are posted from 1,912 different sources, in which 1,707 different sources are used by normal users and 869 different sources are used by marionette users. Table 1 lists the top 5 mostly used sources for normals and marionettes respectively. We find that more than

half tweets of normal users are posted via "Sina Web", thus the web interface remains the primary choice for accessing microblog and "iPhone" and "Android" are two most popular mobile clients. While for marionette users, most tweets are posted via "Sina Mobile" which denotes that majority tweets are posted via the web browsers of cell phones. In this case, if massive user accounts are created from some mobile IP address, the microblog service could not block this IP as it could be the real requests from normal users in the same district. Besides, the IP address can change when the puppeteer relocates.

Table 1. Top 5 Most Used Devices to Post Tweets

Normal		**Marionette**	
Device	**#Tweet**	**Device**	**#Tweet**
Sina Web	356,192	Sina Mobile	209,739
iPhone	59,996	Sina Web	29,365
Android	54,778	UC Browser	4,775
Sina Mobile	19,733	Android	2,577
S60	19,278	iPhone	2,112

Besides above local features, we also select: *the maximal, minimal, middle and average length of tweets*; *the maximal, minimal, middle and average time interval between tweets*; *the percentage of retweets*. We did not include the word bag features here, this is because we want make the model more generic. Since the marionette users owned by the same backstage puppeteer will retweet the same tweet, if the bag-of-word features are utilized as features, the trained model will incline to these word features and become over fit. To our knowledge, the bot detection of a popular search engine [5] only use behavior features, the words are used in blacklist for pre-filtering.

3.3 Classification Evaluation

To show the advantages of incorporating social relations, we compare with the baseline method which only applies Logistic Regression on the local features without considering social relations. When evaluating the proposed model, we set different values of d and different numbers of neighbors to illustrate the impact of social relations on the marionette user detection task. We implement the proposed method based on Weka [6] and the recorded accuracy is the average computed based on 5-fold cross validation.

- Baseline: The baseline model is a Logistic Regression classification model which adopts the local features introduced in previous sub section.
- Light-Neighbor: This model is the proposed model which adopts the same local features as the baseline model and incorporates the social relations with the setting of 5 neighbors and $d = 0.1$.

– Heavy-Neighbor: Similar to Light-Neighbor model, except this model biases more towards social relations with a higher degree setting $d = 0.5$.

Table 2. Classification Results on Three Models

	Precision	Recall	F-Measure
Baseline	0.884	0.875	0.872
Light-Neighbor	0.900	0.890	0.887
Heavy-Neighbor	**0.907**	**0.895**	**0.892**

Table 2 lists the weighted classification precision, recall and f-measure over three models. We can find that incorporating social relation increases the performance of detecting marionette users.

We also evaluate the proposed model with different settings of d and different number of neighbors and show the results in Figure 3. From this figure, we can find that in general when the number of neighbors increases, the classification accuracy increases. Similarly, when the value d increases, the accuracy improves as well. This clearly demonstrates the importance of casting social relations in the classification model. The more neighbors we included and the stronger influence we give to social relations, the better the performance is.

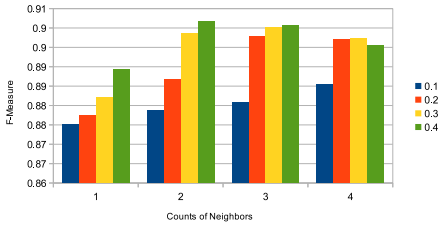


Fig. 3. Classification F-Measure with Different Neighbor and Degree Settings

3.4 Credibility Measurement

We further apply the proposed probabilistic model to detect the credibility of hot retweets. Firstly we apply the model learned from the *Classification Corpus* to classify the users in *Suspicious Hot Tweets Corpus*, and then we can obtain the percentage of marionette users who retweet the hot tweets.

Table 3 lists some Weibo accounts that post suspicious hot tweets, the possible promotion purpose and the percentage of marionette users. The percentages for the first four tweets are quite high, which suggests that most of their retweets are conducted by marionette users. Although the retweet of the last tweet shown

in Table 3 involves more normal users, it might be attributed to the fact that marionette users attract the attention of many normal users and thus the goal of promotion is achieved through marionette user purchase.

Table 3. The Marionette User Percentage of Suspicious Hot Tweets

Tweet Author	Promotion Purpose	Marionette
A Web Site of Clothing Industry	Web Site Promotion	100.00%
A Famous Brand of Women's Dress	Weibo Account Promotion	98.61%
Provincial Culture Communication Co., LTD	Ceremony Advertisement	93.62%
A Anti-Worm Software for Mobile Device	A Security Issue Reminder	92.44%
A Famous China Smart Phone Manufacturer	The Advertisement of Sale Promotion	43.04%

4 Related Works

In this section, we describe related works from two perspectives: (1) the credibility issues of web data and corresponding solutions; (2) the credibility issues of microblog data.

4.1 Credibility of Web Data

Prior to the emergence of microblog services, the web has existed for over two decades. Many web services have been experiencing all kinds of malicious attacks. For example, the robot users submit specific queries or conduct fake clicks towards search engines, aiming to hack the ranking or auto-suggestion results [7]. The approaches like [5, 8] have been proposed to detect and exclude such automated traffic. Different from the robot users, the marionette users possess social relations which can be utilized to build better classifiers.

Besides robot users and automated traffic, another web data issue is the link spam that tries to increase the PageRank of certain pages by creating a large number of links pointing to them. [9–12] propose to optimize search engine ranking and minimize the effects of the link spam. The marionette user detection is different from link spam detection, as the former is a classification problem which targets to separate the marionette users from normal users, while the latter is a ranking problem that targets to lower the rank of link spam web pages. Moreover, the link spam detection methods like [9] rely on the large link structure on the web, while the marionette detection only requires the local features and social connections of each user.

4.2 Credibility of Microblog Data

Due to the massive usage of microblog data, the credibility of microblog data becomes extremely important. [13] explored the information credibility of news propagated through Twitter and proposed to assess the credibility level of newsworthy topics. [14, 15] identified the "Link Farmer" in microblog systems. This

type of users try to acquire more followers and distribute spams. The main difference between the link farmers and marionette users is that the former one is seeking for followers and the latter one is providing followers. [16, 17] analyzed the possible harm that link farmers could have done to microblog applications and [18] proposed several classifiers to detect the link farmers on Facebook. [19] identified the cyber criminals. Different from marionette users, the cyber criminals generate direct harm to normal users by spreading phishing scams.

The SMFSR method proposed by [20] is related to the proposed approach in the sense that it combines user activities, social regularization and semi-supervised labeling in one framework. Specifically, it employed a matrix factorization based method to find spammers in social networks. Different from the proposed approach, this method is transductive rather than inductive. In other words, it is difficult to be used tof predict over new users not originally in the training set. Every time, new users are added, the entire matrix factorization needs to run again.

5 Conclusions

In the paper, we first discuss the business model of puppeteers and marionette users or how they make profits in microblog services. The following facts motivate the emergence of marionette user purchase: 1) to increase the number of followers and fake their popularity, some users purchase marionette users to follow them; and 2) to increase the retweet time and make promotion tweet to the front page story, the advertiser pays marionette users to retweet their tweets. Marionette users cheat in microblog services by manipulating fake retweets and following relations. Therefore, to ensure information trustworthiness and security guarantee, it is extremely important to detect marionette users in a timely manner. Facing the challenges posed by the complicated strategies adopted by marionette users, we propose an effective probabilistic model to fully utilize local user features and social relations in detecting marionette users. We propose an iterative EM procedure to infer model parameters from data and the model can then be used to predict whether a user is marionette or normal. Experiments on Sina Weibo data show that the proposed method achieves a very high f-measure close to 0.9, and the further analysis on some retweet examples demonstrates the effectiveness of the proposed model in measuring the true credibility of information on microblog platforms.

References

1. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World wide web. WWW '10 (2010) 851–860
2. Yu, L., Asur, S., Huberman, B.A.: Artificial inflation: The real story of trends in sina weibo. http://www.hpl.hp.com/research/scl/papers /china-trends/weibospam.pdf (2012)

3. Bollen, J., Mao, H., Zeng, X.J.: Twitter mood predicts the stock market. CoRR **abs/1010.3003** (2010)
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. J. Artif. Int. Res. **16**(1) (June 2002) 321–357
5. Kang, H., Wang, K., Soukal, D., Behr, F., Zheng, Z.: Large-scale bot detection for search engines. In: Proceedings of the 19th international conference on World wide web. WWW '10 (2010) 501–510
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. **11**(1) (2009) 10–18
7. Buehrer, G., Stokes, J.W., Chellapilla, K.: A large-scale study of automated web search traffic. In: Proceedings of the 4th international workshop on Adversarial information retrieval on the web. AIRWeb '08, ACM (2008) 1–8
8. Yu, F., Xie, Y., Ke, Q.: Sbotminer: large scale search bot detection. In: Proceedings of the third ACM international conference on Web search and data mining. WSDM '10 (2010)
9. Gyöngyi, Z., Garcia-Molina, H., Pedersen, J.: Combating web spam with trustrank. In: Proceedings of the Thirtieth international conference on Very large data bases - Volume 30. VLDB '04, VLDB Endowment (2004) 576–587
10. Wu, B., Davison, B.D.: Identifying link farm spam pages. In: Special interest tracks and posters of the 14th international conference on World Wide Web. WWW '05 (2005) 820–829
11. Krishnan, V., Raj, R.: Web spam detection with anti-trust rank. In: AIRWeb'06. (2006) 37–40
12. Benczur, A.A., Csalogany, K., Sarlos, T., Uher, M., Uher, M.: Spamrank - fully automatic link spam detection. In: In Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb). (2005)
13. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on twitter. In: Proceedings of the 20th international conference on World wide web. WWW '11 (2011) 675–684
14. Ghosh, S., Viswanath, B., Kooti, F., Sharma, N.K., Korlam, G., Benevenuto, F., Ganguly, N., Gummadi, K.P.: Understanding and combating link farming in the twitter social network. In: Proceedings of the 21st international conference on World Wide Web. WWW '12 (2012) 61–70
15. Wagner, C., Mitter, S., Körner, C., Strohmaier, M.: When social bots attack: Modeling susceptibility of users in online social networks. In: 2nd workshop on Making Sense of Microposts at WWW2012. (2012)
16. Silvia Mitter, C.W., Strohmaier, M.: Understanding the impact of socialbot attacks in online social networks. In: WebSci. (2013) 15–23
17. Boshmaf, Y., Muslukhov, I., Beznosov, K., Ripeanu, M.: Design and analysis of a social botnet. Comput. Netw. **57**(2) (February 2013) 556–578
18. Reddy, R.N., Kumar, N.: Automatic detection of fake profiles in online social networks. http://ethesis.nitrkl.ac.in/3578/1/thesis.pdf (2012)
19. Yang, C., Harkreader, R., Zhang, J., Shin, S., Gu, G.: 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. WWW '12 (2012) 71–80
20. Zhu, Y., Wang, X., Zhong, E., Liu, N.N., Li, H., Yang, Q.: Discovering spammers in social networks. In: AAAI. (2012)