Detecting Spam Bots in Online Social Networking Sites: A Machine Learning Approach

Alex Hai Wang

College of Information Sciences and Technology,
The Pennsylvania State University,
Dunmore, PA 18512, USA
hwang@psu.edu

Abstract. As online social networking sites become more and more popular, they have also attracted the attentions of the spammers. In this paper, Twitter, a popular micro-blogging service, is studied as an example of spam bots detection in online social networking sites. A machine learning approach is proposed to distinguish the spam bots from normal ones. To facilitate the spam bots detection, three graph-based features, such as the number of friends and the number of followers, are extracted to explore the unique follower and friend relationships among users on Twitter. Three content-based features are also extracted from user's most recent 20 tweets. A real data set is collected from Twitter's public available information using two different methods. Evaluation experiments show that the detection system is efficient and accurate to identify spam bots in Twitter.

1 Introduction

Online social networking sites are becoming more popular each day, such as Facebook, Twitter, and LinkedIn. Among all these sites, Twitter is the fastest growing one than any other social network, surging more than 2,800% in 2009 according to Opera's State of the Mobile Web report [1]. Unfortunately, spam is becoming an increasing problem on Twitter as other online social network sites.

Spammers use Twitter as a tool to post multiple duplicate updates containing malicious links, abuse the reply function to post unsolicited messages to users, and hijack trending topics. Spammers also pushed offensive terms on to Twitter *trending topics*, which displays on Twitter's front page, for several times. This forced Twitter to temporarily disable the trending topic and remove the offensive terms.

Twitter tried several ways to fight spam, which include adding a "report as spam" feature to its service and cleaning up suspicious accounts. However, legitimate Twitter users complain that their accounts are getting caught up in Twitters anti-spam actions. Twitter recently admitted to accidentally suspending accounts as a result of a spam clean-up effort.

In this paper, the suspicious behaviors of spam bots are studied. My goal is to apply machine learning methods to distinguish spam bots from normal ones.

The rest of the paper is organized as follow. In Section 2, the related works are discussed. In Section 3, novel content-based and graph-based features are proposed to facilitate spam bots detection. Bayesian classification method is applied in Section 4 to detect spam bots in Twitter. Section 5 introduces the two data set collecting methods. Experiments are also conducted to evaluate the performance of detection system.

2 Related Work

Spam detection has been studied for a long time. The previous work mainly focuses on email spam detection and Web spam detection. In [2], Sahami et al. first proposed a Bayesian approach to filter spam emails. Experiment results show that the classifier has a better performance considering domain-specific features in addition to the raw text of E-mail messages. Currently spam email filtering is a fairly mature technique. Bayesian spam email filters are implemented both on modern email clients and servers.

Not like email system, Web is massive, changes more rapidly, and is spread over geographically distributed computers [3]. It is a significant challenge to detect Web spam. [4] first formalized the problem and proposed a comprehensive solution to detect Web spam. The TrustRank algorithm is proposed to compute the trust score for a Web graph. Based on computed scores where good pages are given higher scores, spam pages can be filtered in the search engine results. In [5], the authors based on the link structure of the Web proposed a measurement Spam Mass to identify link spamming. A directed graph model of the Web is proposed in [6]. The authors apply classification algorithms for directed graphs to detect real-world link spam. In [7], both link-based features and content-based features are proposed. The basic decision tree classifier is implemented to classify spam. In [8], semi-supervised learning algorithms are proposed to boost the performance of a classifier which only needs small amount of labeled samples.

For spam detection in other applications, Wu et al. [9] present an approach for detection of spam calls over IP telephony called SPIT in VoIP system. Based on the popular semi-supervised learning methods, a improved algorithm called MPCK-Means is proposed. In [10], the authors study the video spammers and promoters detection in YouTube. By far this is the only work I found studying spam detection in online social network sites.

In [11], the authors collected three datasets of the Twitter network. The Twitter users' behaviors, geographic growth pattern, and current size of the network are studied.

3 Features

The features extracted for spam detection include three graph-based features and three content-based features. As a social networking site, Twitter allows users to build their own social graph. Three graph-based features are extracted from Twitter's social graph to capture the "*following*" relationship among users.

Twitter also allows users to broadcast short messages in 140 characters, known as "*tweet*", to their friends or followers. I extract three content-based features from the user's 20 most recent tweets.

3.1 Graph-based Features

Following is one of the most important and unique functions about Twitter. Users can build their own social network by following friends and allowing others to follow them on Twitter. You can follow your friends' accounts to get their updates automatically on your Twitter homepage when you log in. And your friends can send your private messages, called *direct messages*, if you follow them. Spammers use the following function to take legitimate users' attention by following their accounts, since Twitter will send an email notification when someone follows your account. Twitter considers it as a spam bot, if this account "has small number of followers compared to the amount of people you are following".

Three graph-based features, which are the number of friends, the number of followers, and the follower ratio, are extracted to detect spam bots on Twitter. If someone follows your account, it becomes one of your followers. If you follow someone's account, then it becomes one of your friends. The number of friends and the number of followers are extracted for each individual Twitter account.

Furthermore, the follower ratio is computed based the number of followers and the number of to friends. Let N_{fo} denote the number of followers, N_{fr} denote the number of friends, and r_{ff} denote the follower ratio. To normalize the follower ratio, this feature is defined as the ratio between the number of people you are following and the number of people following you.

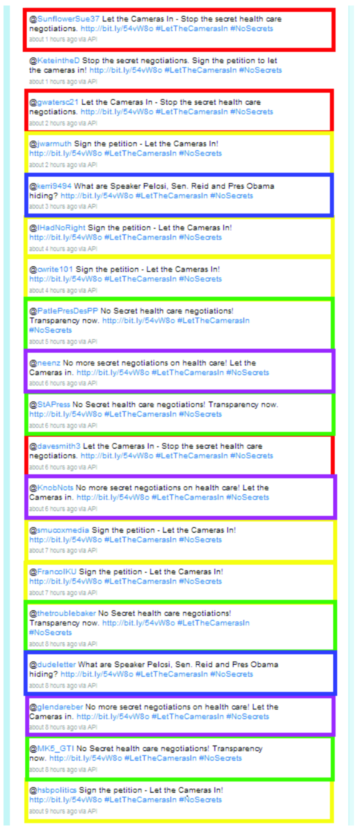
$$r_{ff} = \frac{N_{fo}}{N_{fo} + N_{fr}} \tag{1}$$

Obviously if the number of followers is relatively small compared to the amount of people you are following, the follower ratio is relatively small and close to zero. At the same time the probability that the associated account is spam is high.

3.2 Content-based features

In this part, novel content-based features extracted from Twitter are introduced. Three features, which are the number of duplicate tweets, the number of HTTP links, and the number of replies/mentions, are extracted from the user's 20 most recent tweets.

First, an account may be considered as a spam if it posts duplicate content on one account. A sample Twitter spam page is shown in Figure 1. Usually legitimate users will not post duplicate updates. Duplicate tweets are detected by measuring the Levenshtein distance (also known as edit distance) between two different tweets posted by the same account. The Levenshtein distance is defined as the minimum cost of transforming one string into another through a sequence of edit operations, including the deletion, insertion, and substitution of individual symbols. The distance is zero if and only if the two tweets are identical.



@SunflowerSue37 Let the Cameras In - Stop the secret health care negotiations. http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 1 hours ago via API

@KeteintheD Stop the secret negotiations. Sign the petition to let the cameras in! http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 1 hours ago via API

@gwatersc21 Let the Cameras In - Stop the secret health care negotiations. http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 2 hours ago via API

@jwarmuth Sign the petition - Let the Cameras In! http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 2 hours ago via API

@kerri9494 What are Speaker Pelosi, Sen. Reid and Pres Obama hiding? http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 3 hours ago via API

@IHadNoRight Sign the petition - Let the Cameras In! http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 4 hours ago via API

@cwrite101 Sign the petition - Let the Cameras In! http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 4 hours ago via API

@PatlePresDesPP No Secret health care negotiations! Transparency now. http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 5 hours ago via API

@neenz No more secret negotiations on health care! Let the Cameras in. http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 6 hours ago via API

@StAPress No Secret health care negotiations! Transparency now. http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 6 hours ago via API

@davesmith3 Let the Cameras In - Stop the secret health care negotiations. http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 6 hours ago via API

@KnobNots No more secret negotiations on health care! Let the Cameras in. http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 6 hours ago via API

@smucoxmedia Sign the petition - Let the Cameras In! http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 7 hours ago via API

@FrancoIKU Sign the petition - Let the Cameras In! http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 7 hours ago via API

@thetroublebaker No Secret health care negotiations! Transparency now. http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 8 hours ago via API

@dudeletter What are Speaker Pelosi, Sen. Reid and Pres Obama hiding? http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 8 hours ago via API

@glendareber No more secret negotiations on health care! Let the Cameras in. http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 8 hours ago via API

@MK5_GTI No Secret health care negotiations! Transparency now. http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 8 hours ago via API

@hsbpolitics Sign the petition - Let the Cameras In! http://bit.ly/54vW8o #LetTheCamerasIn #NoSecrets
about 9 hours ago via API

Fig. 1. A Twitter spam page (Duplicate tweets are circled in the same color rectangles)

To avoid detection and spam different accounts, spam bots often include different *@usernames* in their duplicates tweets. When the Levenshtein distances are computed between different tweets, I clean the data by deleting *@replies*, *#topic*, and HTTP slinks. In other words, the reply/mention, topic, and link information are ignored when I capture the duplicate tweet, instead only the content of the tweets is considereds.

Second, spam bots try to post malicious links in their tweets to allure legitimate users to click. Since Twitter only allows you to post a message within 140 characters, some URL shortening services and applications, such as bit.ly, become popular to meet the requirements. A shorten URL obscures the target address, and as a result it facilitates the spam accounts in pranks, phishing, or affiliate hiding. So Twitter considers it as a factor of spam if your tweets consist mainly of links, and not personal updates.

The number of links in one account is measured by the number of tweets containing HTTP links in the user's 20 most recent tweets. If a tweet contains the sequence of characters "http://" or "www.", this tweet is considered as containing a link.

Third, the number of replies/mentions is extracted from the user's 20 most recent tweets. On Twitter, users can use the *@+username+message* format to designate their message as a reply to another person . You can reply to anyone's tweet on Twitter no matter they are your friends or not. You can also mention another user name (*@username*) anywhere in the tweet, rather than just the beginning. Twitter collects all tweets containing your user name in the *@username* format in your replies tab. You can see all replies made to you, and mentions of your user name.

The reply and mention functions are designed to help users to discover each other on Twitter. However, the spam account utilize the service to draw other user's attention by sending unsolicited replies and mentions. Twitter also considers this as a factor to determine spamming. The number replies and mentions in one account is measured by the number of tweets containing the reply sign "@" in the user's 20 most recent tweets.

4 Spam Bots Detection

In this section, I apply different classification methods, such as decision tree, neural network, support vector machines, and k-nearest neighbors, to identify spam bots on Twitter. Among these algorithms, Bayesian classifier has the best performance for several reasons. First, Bayesian classifier is noise robust. Another reason that Bayesian classifier has a better performance is that the class label is predicted based on user's specific pattern. A spam probability is calculated for each individual user based its behaviors, instead of giving a general rule. Also, Bayesian classifier is a simple and very efficient classification algorithm.

The Bayesian classifier is based on the well-known Bayes theorem:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{2}$$

The conditional probability of $P(Y|X)$ is also known as the posterior probability for Y, as opposed to its prior probability $P(Y)$.

Each Twitter account is considered as a vector X with feature values as discussed in Section 3. I classify the vectors into two classes Y: spam and non-spam. To classify a data record, the posterior probability is computed for each class:

$$P(Y|X) = \frac{P(Y)\prod_{i=1}^{d} P(X_i|Y)}{P(X)} \tag{3}$$

Since $P(X)$ is a normalizing factor which is equal for all classes, we need only maximize the numerator $P(Y)\prod_{i=1}^{d} P(X_i|Y)$ in order to do the classification.

5 Experiments

5.1 Data Set

The data set is collected using two methods. First I use Twitter's API methods to collect user's detailed information. Second, a Web crawler is developed to extra a specific unauthorized user's 20 most recent tweets.

First I use the *public_timeline* API method to collect information about the 20 non-protected users who have set a custom user icon in real time. This method can randomly pick 20 non-protected users who updated their status recently on Twitter. I extract details of the current user, such as IDs, screen name, location, and etc. At the same time, I also use *social graph* API methods to collect information about user's friends and followers, such as the number of friends, the number of followers, list of friend IDs, list of follower IDs, and etc. The Twitter's *friends* and *followers* APIs can return maximum 5,000 users. If a user has more than 5,000 friends or followers, I could only extract a partial list of friends or followers. Based on my observation, the number of friends and followers of most users do not exceed 5,000, so this constraint does not affect my method significantly.

Another constraint of Twitter API methods is the number of queries per hour. Currently the rate limit for calls to the API is 150 requests per hour. To collect data from different time periods and avoid congesting Twitter Web servers, I crawl Twitter continuously and limit my request 120 calls per hour.

Although Twitter provides neat API methods for us, there is no method that allows us to collect a specific unauthorized user's recent tweets. The *public_timeline* API method can only return the most recent update from 20 different non-protected users (one update from one user). The *user_timeline* API method can return the 20 most recent tweets posted only from an authenticating user. The recent tweets posted by a user are important to extract content-based features, such as duplicate tweets. To solve this problem, I develop a web crawler to collect the 20 most recent tweets of a specific non-protected user based on the user's ID on Twitter. The extracted tweets are saved both as a XML file and into a relational database.

Finally, I collect the data set for 3 weeks from January 3 to January 24, 2010. Totally 25,847 users, around 500K tweets, and around 49M follower/friend relationships are collected from the public available data on Twitter.

5.2 Evaluation

To evaluate my method, I manually labeled 500 Twitter user accounts to two classes: spam and not spam. Each user account is manually evaluated by reading the 20 most recent tweets posted by the user and checking the friends and followers of the user. The result shows that there is around 1% spam account in the data set. The study [12] shows that there is probably 3% spam on Twitter. To simulate the reality and avoid the bias in my crawling method, I add more spam data to the data set.

As mentioned in Section 1, Twitter provides several method for users to report spam, which includes sending Twitter a direct message and clicking on the "report for spam" link. The most simple and public available method is to post a tweet in the "*@spam @username*" format where *@username* is to mention the spam account. I queried "*@spam*" to collect additional spam data set. Surprisedly I found that this service is abused by hoaxes and spam. Only a small percentage of *@spam* tweets is reporting spam. I clean the query results by manually evaluating each spam report. Finally the data set is mixed of containing around 3% spam.

The evaluation of the overall process is based on a set of measures commonly used in Machine Learning and Information Retrieval. Given a classification algorithm C, I consider its confusion matrix:

		Prediction	
		Spam	Not Spam
True	Spam	a	b
	Not Spam	c	d

Three measures are considered in the evaluation experiements: precision, recall, and F-measure. The precision is $P = a/(a + c)$ and the recall is $R = a/(a + b)$. The F-measure is defined as $F = 2PR/(P + R)$. For evaluating the classification algorithms, I focus on the F-measure F as it is a standard way of summarizing both precision and recall.

All the predictions reported in the paper are computed using 10-fold cross validation. For each classifier, the precision, recall, and F-measure are reported. Each classifier is trained 10 times, each time using the 9 out of the 10 partitions as training data and computing the confusion matrix using the tenth partition as test data. The evaluation metrics are estimated on the average confusion matrix. The evaluation results are shown in Table 1. The naïve Bayesian classifier has the best overall performance compared with other algorithms.

Table 1. Classification Evaluation

Classifier	Precision	Recall	F-measure
Decision Tree	0.667	0.333	0.444
Neural Networks	1	0.417	0.588
Support Vector Machines	1	0.25	0.4
Naïve Bayesian	0.917	0.917	0.917

6 Conclusion

In this paper, I focus on the suspicious behaviors of spam bots in online social networking sites. A popular micro-blogging service, called Twitter, is studied as an example. A machine learning approach is proposed to identify the spam bots from normal noes. Based on the spam policy on Twitter, graph-based features and content-based features are extracted from user's social graph and most recent tweets. Traditional classification algorithms are applied to detect suspicious behaviors of spam bots. A Web crawler using Twitter API is developed to collect real data set from Twitter public available information. Finally, I analyze the data set and evaluate the performance of the detection system. Several popular classification algorithms are studied and evaluated. The results show that the Bayesian classifier has a better overall performance.

References

1. Opera: State of the mobile web. http://www.opera.com/smw/2009/12/
2. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A bayesian approach to filtering junk e-mail. In: AAAI Workshop on Learning for Text Categorization. (1998)
3. Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., Raghavan, S.: Searching the web. ACM Trans. Internet Technol. **1**(1) (2001) 2–43
4. Gyöngyi, Z., Garcia-Molina, H., Pedersen, J.: Combating web spam with trustrank. In: Proceedings of the Thirtieth international conference on Very large data bases. (2004) 576–587
5. Gyongyi, Z., Berkhin, P., Garcia-Molina, H., Pedersen, J.: Link spam detection based on mass estimation. In: VLDB '06: Proceedings of the 32nd international conference on Very large data bases. (2006) 439–450
6. Zhou, D., Burges, C.J.C., Tao, T.: Transductive link spam detection. In: Proceedings of the 3rd international workshop on Adversarial information retrieval on the web. (2007) 21–28
7. Castillo, C., Donato, D., Gionis, A., Murdock, V., Silvestri, F.: Know your neighbors: web spam detection using the web topology. In: Proceedings of the 30th annual international ACM SIGIR conference. (2007) 423–430
8. Geng, G.G., Li, Q., Zhang, X.: Link based small sample learning for web spam detection. In: Proceedings of the 18th international conference on World wide web. (2009) 1185–1186
9. Yu-Sung Wu, Bagchi, S.S.N.W.R.: Spam detection in voice-over-ip calls through semi-supervised clustering. In: Proceedings of the 2009 Dependable Systems Networks. (2009) 307 –316
10. Benevenuto, F., Rodrigues, T., Almeida, V., Almeida, J., Gonçalves, M.: Detecting spammers and content promoters in online video social networks. In: Proceedings of the 32nd international ACM SIGIR conference. (2009) 620–627
11. Krishnamurthy, B., Gill, P., Arlitt, M.: A few chirps about twitter. In: WOSP '08: Proceedings of the first workshop on Online social networks. (2008) 19–24
12. Analytics, I.P.: Twitter study. http://www.pearanalytics.com/wp-content/uploads/2009/08/Twitter-Study-August-2009.pdf