UNIVERSITY OF WINDSOR, SCHOOL OF COMPUTER SCIENCE

Related Works of Detecting Spammers on Twitter

Xiutian Cui

June 7, 2015

1 DETECTING SPAMMERS ON TWITTER

Benevenuto et al. [2] addressed a study on the spammers who focused on sending spam concluding the trending topics in Twitter in 2010. The main method they used is to collect user profiles and tweets, then classify them into two groups, spammer and non-spammer, by using Support Vector Machine (SVM). There are four steps in their approach, crawling user data, labeling users, analyzing the characteristics of tweet content and user behaviours and using a supervised classifier to identify spammers.

1.1 STUDY APPROACH

1.1.1 CRAWLING TWITTER

The authors collected all user IDs ranging from 0 to 80 million since August 2009, which have been considered as all users on Twitter since there is no single user in the collected data had a link to one user whose ID is greater than 80 million. Finally they collected 54,981,152 used accounts that were connected to each other by 1,963,263,821 social links, together with 1,755,925,520 tweets. Among those users, there are 8% accounts were set private and were ignored. The detail description of this dataset can be found on their project homepage[1].

1.1.2 BUILDING A LABELED COLLECTION

In order to classify the users into spammers and non-spammers, the authors used supervised classifier. So they need to label one collection that contains spammers and non-spammers. In this paper they focused on the users who sent the tweets about trending topic, so they

need to build one collection of users who sent topics of (1) the Michael Jackson's death, (2) Susan Boyle's emergence, and (3) the hashtag "#musicmonday". 8,207 users have been labeled manually, including 355 spammers and 7,852 non-spammers. They then randomly chose 710 non-spammers to reduce the number of non-spammers. Thus, the total size of labeled collection is 1,065 users.

1.1.3 IDENTIFYING USER ATTRIBUTES

To use machine learning algorithms, the authors then identified the attributes of users. The attributes are divided into two categories: content attributes and user behavior attributes. Content attributes are the ones represented in what the users posted. User behavior attributes are the properties of the users' acting on Twitter. Both of these two kinds of attributes are shown in Table 1.1.

1.2 EXPERIMENTS

The authors used SVM to classify user collections with the attributes that they identified in the previous section. The implementation of SVM they used in their experiments is provided by libSVM. Each user in the user collection is presented by a vector of values, which contains the attributes of this user. SVM will first trains a model from the labeled user dataset, and then applies this model to the classify the unknown users into two classes: spammers and non-spammers.

Table 1.2 shows the confusion matrix of classification result. About 70% of spammers and 96% of non-spammers were correctly classified. The Micro-F1 (which is calculated by first computing global precision and recall values for all classes, and then calculating F1) is 87.6, indicating that 87.6% of the cases were correctly predicted.

To reduce the misclassifying of non-spammers, the authors used two approaches. First is to adjust J parameter in SVM. In SVM, J parameter can be used to give priority to one class over the other. With the varying of J, the rate of correctness of classify can be increased to 81.3% ($J = 5$), with the misclassifying of legitimate users has been increased to 17.9%.

The second approach they used is to reduce the size of attributes set. By sorting the attributes by their importance, the authors can remove the non-important attribute and give more weight to the important ones. They used two feature selection methods, information gain and χ^2, which are available in Weka. The results of these two methods are similar and the top 10 attributes in result are same. Table 1.3 shows the top 10 result of feature selection. And the result of classification when just using top 10 attributes instead of all attributes shows that top 10 attributes are enough to classify the users.

Category	Attribute
Content Attributes	number of hashtags per number of words on each tweet
	number of URLs per words
	number of words of each tweet
	number of characters of each tweet
	number of URLs on each tweet
	number of hashtags on each tweet
	number of numeric characters (i.e. 1,2,3) that appear on the text, number of users mentioned on each tweet
	number of times the tweet has been retweeted (counted by the presence of "RT @username" on the text)
User Behavior Attributes	number of followers
	number of followees
	fraction of followers per followees
	age of the user account
	number of times the user was mentioned
	number of times the user was replied to
	number of times the user replied someone
	number of followees of the userâĂŹs followers
	number tweets receveid from followees
	existence of spam words on the userâĂŹs screename
	the minimum, maximum, average, and median of the time between tweets
	number of tweets posted per day and per week

Table 1.1: User Attributes

		Predicted	
		Spammer	Non-spammers
True	Spammer	70.1%	29.9%
	Non-spammer	3.6%	96.4%

Table 1.2: Basic classification result

Rank	Attribute
1	fraction of tweets with URLs
2	age of the user account
3	average number of URLs per tweet
4	fraction of followers per followees
5	fraction of tweets the user had replied
6	number of tweets the user replied
7	number of tweets the user receive a reply
8	number of followees
9	number of followers
10	average number of hashtags per tweet

Table 1.3: Basic classification result

1.3 Conclusion

In this paper, the authors proposed an approach to detecting spammers on Twitter. The results of their experiments showed that their approach is able to detect spammers with high accuracy. Although results for this approach showed to be competitive, the spammer classification uses a much larger set of attributes and is more robust to spammers that adapt their spamming strategies.

2 Can You Judge a Man by His Friends? - Enhancing Spammer Detection on the Twitter Microblogging Platform Using Friends and Followers

Moh et al. [6] analyzed how much information gained from the friends and followers of one user in 2010. They also proposed a learning process to determine whether or not a user is spammer. There are two steps in this process. The first step is to train a categorization algorithm to distinguish between spammers and non-spammers on a set of basic user features. And the second step is to train a classifier to generate new features, which depend on a user's followers being spammers or non-spammers.

2.1 Spam Categorization Framework

The outline of spam categorization frame that the authors introduced is as follows:

Data: $trainingSet$, $testSet$
$model1 \leftarrow$ TRAIN-BASIC-LEARNER($trainingSet$)
$trainSetEx \leftarrow$ EXTEND-DATA($trainingSet$, $model1$)
$model2 \leftarrow$ TRAIN-BASIC-LEARNER($trainSetEx$)
for $e \in testSet$ **do**
    $extended \leftarrow$ EXTEND-DATA(e, $model1$)
    EVALUATE($extended$, $model2$)
end

Algorithm 1: ENHANCED ATTRIBUTE LEARNER

The first step of this learner is to train a model based on manually labeled user collections. And then one extended attribute set will be generated for each user based on the predictions provided by the first learner and the user's position in the social network. The learner will then be trained on this extended attribute set.

2.2 Experiments

2.2.1 Test Data

All the user data are from Twitter. Most of the account names of spammers were acquired using the web page twitspam.org, where users can submit the names of suspected spammers. Another part of spammers are added by the authors during they collected data. They obtained non-spammers from the users they followed. In total they collected one dataset that contains 77 spammers and 155 non-spammers. And for each user in this dataset, they also collected the information on up to 200 of their followers.

2.2.2 Approach

The authors built up two attribute set for each user. The basic attribute set here contains:

- follower-friend ratio
- number of posts marked as favorites
- friends added per day
- followers added per day
- account is protected?
- updates per day
- has url?
- number of digits in account name
- reciprocity.

The authors added new two features here: number of digits in account name, and reciprocity. The number of digits in account name has been proved useful in classification by Krause et al.[5] And the reciprocity, which is the rate of one user follows his follower, has been added because the spammers may try to follow those who will follow their followers.

The aggregating friend and follower (peer) attribute set contains:

- follower-friend ratio
- updates per day
- friends added per day
- followers added per day
- reciprocity
- account is protected?

These features were based on attributes of a user's peers. Each attribute in this attribute set was calculated separately for all followers and friends of a user.

The second step is to compute trust metric. The authors modified the original formula.

$$\text{trust metric} = \sum_{followers} \frac{1}{\#\text{users followed}}$$

They applied the following modifiers to this formula:

- **legit** accumulate only the values coming from users who are predicted to be legitimate users
- **capped** accumulate only values coming from up to 200 users
- **squared** use $\frac{1}{\#\text{users followed} \times \#\text{users followed}}$ instead of $\frac{1}{\#\text{users followed}}$

2.2.3 Results

For there are two steps in classification, the authors tried different combination of classifiers. Then they calculated the accuracy, precision, recall, F1, and finally draw a Receiver Operating Characteristic Curve (ROC curve) to evaluate the test results of each combination. Table 2.1 and Table 2.2 show the evaluation metrics for RIPPER algorithm and C4.5 algorithm.

Metric	basic	basic+peer	peer	basic+trust	all features
Precision	0.79	0.80	0.75	0.88	0.84
Recall	0.84	0.83	0.71	0.85	0.85
F1	0.81	0.81	0.73	0.87	0.84
Accuracy	0.87	0.87	0.82	0.91	0.90

Table 2.1: Evaluation metrics for RIPPER algorithms with the different extended feature sets

Metric	basic	basic+peer	peer	basic+trust	all features
Precision	0.80	0.81	0.72	0.85	0.86
Recall	0.85	0.79	0.67	0.85	0.86
F1	0.83	0.80	0.69	0.85	0.86
Accuracy	0.88	0.87	0.80	0.90	0.90

Table 2.2: Evaluation metrics for C4.5 algorithms with the different extended feature sets

The authors also tried to measure the information provided by each features. To do so, the authors calculated the information gain and the chi square values for each feature in extended feature set.

The authors claimed that using RIPPER in two steps achieved the best performance among the combinations of classifiers. And top 10 features ranked by information gain and chi square value is shown in Table. 2.3 and Table. 2.4.

Attribute	Chi square value
spammers to legit followers	128.68
friends per day	106.72
trust metric legit.	105.49
friend-follower ratio	101.23
trust metric legit. capped	94.8697
trust metric	88.78
friend-follower average for friends	81.54
average protected for followers	80.57
trust metric legit. square	79.93
trust metric legit. square capped	74.99

Table 2.3: Top 10 Chi Square values for data set extended with RIPPER. Added attributes are bold.

Attribute	Information gain
spammers to legit followers	0.48
friend-follower ratio	0.35
friends per day	0.34
trust metric legit.	0.34
trust metric legit. capped	0.29
trust metric	0.29
friend-follower average for friends	0.27
average protected for followers	0.25
trust metric legit. square	0.24
average protected for friends	0.24

Table 2.4: Top 10 Information gain values for data set extended with RIPPER. Added attributes are bold.

2.3 Conclusion

The much improved classification results and the high values received by additional attributes for both the chi-squared statistic and information gain show that a user's peers indeed tell much about the nature of a user. The RIPPER algorithm was able to get consistently better results on the extended attribute sets as compared to the basic attribute set.

3 Understanding and Combating Link Farming in the Twitter Social Network

In 2012, Ghosh et al. [4] analyzed over 40,000 spammer accounts suspended by Twitter and found out that link farming is wide spread and that a majority of spammers' links are farmed from a small fraction of Twitter users, the social capitalists, who are themselves seeking to amass social capital and links by fol- lowing back anyone who follows them. And they proposed a ranking system, *Collusionrank*, to penalize users from connecting to spammers.

3.1 Analysis of Link Farming in Twitter

3.1.1 Dataset Description

The dataset they used includes a complete snapshot of the Twitter network and the complete history of tweets posted by all users as of August 2009 [3]. To identify the spammers in this dataset, they collected the user accounts which are suspended by Twitter. Although the primary reason for suspension of accounts is spam-activity, the accounts which are inactive for more than 6 months can also be suspended. One URL blacklist which contains the most popular URLs in spam tweets has been constructed to confirm that the suspended users are truly spammers. The authors fetched all the bit.ly or tinyurl URLs that were posted by each of the 379,340 suspended accounts and found that 41,352 suspended accounts had posted at least

one shortened URL blacklisted by either of these two shortening services. These suspended accounts were considered to be spammers.

3.1.2 Spammers farm links

The authors studied how spammers acquire links to study link farm in Twitter by analyzing the nodes following and followed by the 41,352 spammers. They defined the nodes followed by a spammer as *spam-targets* and the nodes that follow a spammer as *spam-followers*. Spam-targets who also follow the spammer are called *targeted followers*. After computing the numbers of spammer-targets, spammer-followers and targeted followers, they found out that the majority (82%) of spam-followers have also been targeted by spammers. And targeted followers are likely to reciprocate most links from spammers. Top 100,000 spammer followers (rank based on the number of links they created to the spammers) exhibited a reciprocation of 0.8 on average and created 60% links to the spammers.

The authors also computed the Pagerank of each user in this dataset and found out that by acquired large farm links from spammer followers, some of the rank of spammers are very high - 7 spammers rank within the top 10,000 (0.018% of all users) 304 and 2,131 spammers rank within the top 100,000 (0.18% of all users) and 1 million (1.8% of all users) users according to Pagerank, respectively.

3.1.3 Analysis of Link Farmers

The authors then analyzed the users who willing to reciprocate links from arbitrary users and the reason why they need to farm links. They plotted how the probability of a user reciprocating to a link from spammers varies with the user's indegree and found out that the lay users, who have low indegree, rarely respond to spammers. On the other hand, users with high indegree value are more likely to follower a spammer.

And the authors also found out that the top link farmers (top 100,000 spam-followers) sometimes are active contributors instead of spammers. The motivating factor for such users might be the desire to acquire social capital and thereby, influence.

3.2 Collusionrank

The authors proposed Collusionrank, a Pagerank-like approach, to combat link farming in Twitter. Algorithm 2 explains their approach.

Data: network, G; set of known spammers, S; decay factor for biased Pagerank, α
Result: Collusionrank scores, c
initialize score vector d for all nodes n in G

$$d(n) \leftarrow \begin{cases} \frac{-1}{|S|} & \text{if} n \in S \\ 0 & \text{otherwise} \end{cases}$$

$c \leftarrow d$
while *c not converged* **do**
  for $n \in G$ **do**
    $tmp \leftarrow \sum_{nbr \in followings(n)} \frac{c(nbr)}{|followers(nbr)|}$
    $c(n) \leftarrow \alpha \times tmp + (1-\alpha) \times d(n)$
  end
end

Algorithm 2: Collusionrank

Collusionrank algorithm can also be combined with any ranking strategy used to identify reputed users, in order to filter out users who gain high ranks by means of link farming.

3.2.1 Evaluating Collusionrank

To evaluating Collusionrank, the authors computed the Collusionrank scores of all users in the Twitter social network, considering as the set of identified spammers S, a randomly selected subset of 600 out of the 41,352 spammers.

The result of evaluation showed the effect of ranking spammers of Collusionrank is great. While more than 40% of the 41,352 spammers appear within the top 20% positions in Pagerank, 94% of them are demoted to the last 10% positions in Collusionrank. Even when only a small set of 600 known spammers is used, this approach selectively filtered out from the top positions of Pagerank, most of the unidentified spammers and social capitalists who follow a large number of spammers.

References

[1] Twitter dataset homepage. `http://twitter.mpi-sws.org`.

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

[3] Cha, M., Haddadi, H., Benevenuto, F., and Gummadi, P. K. Measuring user influence in twitter: The million follower fallacy. *ICWSM 10*, 10-17 (2010), 30.

[4] Ghosh, S., Viswanath, B., Kooti, F., Sharma, N. K., Korlam, G., Benevenuto, F., Ganguly, N., and Gummadi, K. P. Understanding and combating link farming in the

twitter social network. In *Proceedings of the 21st international conference on World Wide Web* (2012), ACM, pp. 61–70.

[5] KRAUSE, B., SCHMITZ, C., HOTHO, A., AND STUMME, G. The anti-social tagger: detecting spam in social bookmarking systems. In *Proceedings of the 4th international workshop on Adversarial information retrieval on the web* (2008), ACM, pp. 61–68.

[6] MOH, T.-S., AND MURMANN, A. J. Can you judge a man by his friends?-enhancing spammer detection on the twitter microblogging platform using friends and followers. In *Information Systems, Technology and Management*. Springer, 2010, pp. 210–220.