# Spam Filtering in Twitter using Sender-Receiver Relationship

Jonghyuk Song[1], Sangho Lee[1] and Jong Kim[2]

[1] Dept. of CSE, POSTECH, Republic of Korea
{freestar,sangho2}@postech.ac.kr
[2] Div. of ITCE, POSTECH, Republic of Korea
jkim@postech.ac.kr

**Abstract.** Twitter is one of the most visited sites in these days. Twitter spam, however, is constantly increasing. Since Twitter spam is different from traditional spam such as email and blog spam, conventional spam filtering methods are inappropriate to detect it. Thus, many researchers have proposed schemes to detect spammers in Twitter. These schemes are based on the features of spam accounts such as content similarity, age and the ratio of URLs. However, there are two significant problems in using account features to detect spam. First, account features can easily be fabricated by spammers. Second, account features cannot be collected until a number of malicious activities have been done by spammers. This means that spammers will be detected only after they send a number of spam messages. In this paper, we propose a novel spam filtering system that detects spam messages in Twitter. Instead of using account features, we use relation features, such as the distance and connectivity between a message sender and a message receiver, to decide whether the current message is spam or not. Unlike account features, relation features are difficult for spammers to manipulate and can be collected immediately. We collected a large number of spam and non-spam Twitter messages, and then built and compared several classifiers. From our analysis we found that most spam comes from an account that has less relation with a receiver. Also, we show that our scheme is more suitable to detect Twitter spam than the previous schemes

**Key words:** Spam, Spam filtering, Social network, Twitter

## 1 Introduction

Twitter has grown tremendously over the past few years. With sites such as Google, YouTube, and Facebook, Twitter is ranked in the top 10 most visited sites [1]. In February 2009, Twitter was the fastest-growing website with a growth rate of 1,382% [2]. In 2011, people sent about 140 million tweets per day and 460,000 new accounts were created per day [3]. The enormous growth of Twitter allows many users to share their information and communicate with each other. This popularity, however, also attracts spammers.

Spammers have several goals, which are phishing, advertising, or malware distribution. These goals are similar to traditional spam in email or blogs, but Twitter spam is different. Twitter limits the length of each message to less than 140 characters. Because of this limitation, spammers cannot put enough information into each message. To overcome this restriction, spammers usually send a spam containing URLs that are created by URL shortening services. When a user clicks the short URLs, he will be redirected to malicious pages. Since the messages are short and the actual spam content is located on external spam pages, it is difficult to apply traditional spam filtering methods based on text mining to Twitter spam.

Many researchers have proposed methods to detect spammers in Twitter [4–12]. These methods are mostly based on the characteristics of social networks. To find spammers and collect their information, honeypot-based approaches have been proposed [4–6]. These studies created several honey-profiles and waited for spammers' contacts. After collecting spammer's activity, they analyzed the collected data and tried to automatically identify spammers by analyzing spammer's behavior. Other researchers tried to automatically detect spammers based on statistical analysis [7–12]. They also collected a large number of user profiles and manually classified the users into spammers and non-spammers. They conducted a study of the characteristics of user profiles, user behaviors and tweet contents based on the collected data. Finally they trained a classifier to identify spammers using data mining techniques.

Previous work has classified spammers with high accuracy, but two critical limitations exist. First, they used the account features such as tweeting interval, content similarity, age, the number of followings and the number of followers. These account features, however, can be manipulated by spammers. For instance, spammers can post both benign and spam tweets at irregular intervals. They can also create several spam accounts and follow each other to raise their reputation in social networks. Moreover, spammers can use accounts created a long time ago to manipulate the age feature. Secondly, previous work is able to detect spammers only after spam has already been sent to legitimate users because user history data is needed to decide whether a user is a spammer or not. To classify a user, previous methods need to know how a user has been tweeting and what a user has been tweeting. Therefore, there is an inevitable delay between spam account creation and its detection. Because of the delay, previous work has been criticized [13]. Even if spammers are detected and removed, they can still create accounts and then send spam again.

In this paper, we propose a spam filtering method in Twitter. Instead of account features, our study considers the relation features between a message sender and a receiver, which are difficult for spammers to manipulate. We construct directed graphs based on the following and followed relations in Twitter. In the graphs, we measure two relation features: *distance* and *connectivity* between users. The distance is the length of the shortest path and the connectivity is measured by using min-cut and random walk. We investigated the distribution of spam messages according to the distance between users. From the experimen-

tal results, we are able to find that most spam comes from users at a distance of more than three hops from receivers. We have also investigated the min-cut and random walk between normal users, and between spammers and normal users. From the results, we verify that the connectivity between normal users is different from the connectivity between spammers and normal users. Since our system does not rely on user history data, it allows service managers or clients to identify spammers in *real-time*. This means that when a user receives a message from a stranger, our system identifies the sender at once. If the sender is identified as a spammer, the message is filtered.

In summary, the main contributions of this paper are as follows:

- We propose a spam filtering system for Twitter. We classify the messages as spam or benign messages by identifying the sender. Our experiments are performed on Twitter data, but we believe that our system can also be applied in other social networks.
- We propose two relation features, which are distance and connectivity, to identify spammers. These relation features are unique features of social networks and are difficult for spammers to forge or manipulate.
- Our system identifies spammers in real-time, meaning that service managers or clients can classify the messages as benign or spam when a message is being delivered.

We organize the remainder of this paper as follows. In Section 2, we briefly present the background on traditional spam and an overview of Twitter. Section 3 explains the overall processes including graph construction and features we used to identify spam. Section 4 describes the experiments and evaluation results. In Section 5, we discuss a few issues that need more consideration and in Section 6, we conclude the paper.

## 2 Background

Spam appears in email, blogs, Short Message Services (SMS), and Social Networking Sites (SNS). Many researchers have proposed schemes to detect spam. The common feature of spam, as defined by the researchers, is that it is *unsolicited* one [14]. However, it is difficult to decide whether a message is unsolicited in receivers' side. Thus, content filtering methods are widely used [15]. In social networking services such as Twitter, however, content filtering approaches are not effective because spam contains only a few words and URLs. Domain and URL blacklisting techniques have also been proposed to filter spam, but Grier *et al.* showed that the blacklists are too slow to protect users since there is a delay before hostile sites are included in blacklists [16]. Moreover URL shortening services make it more difficult to detect sites in blacklists. Thus, the approach is not effective in Twitter because almost all users use URL shortening services due to limitation of message length. Because of these reasons, traditional spam detection approaches are difficult to apply to Twitter. Therefore, a new approach is needed with a focus on the characteristics of Twitter.

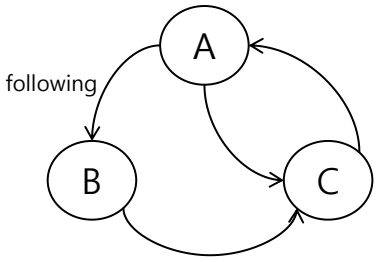



**Fig. 1.** Simple Twitter graph. User A is follower of user B and C and is also following of user C.

### 2.1 Twitter features

There are Twitter-specific features including *tweet*, *mention*, *reply*, *retweet*, *hashtag*, *following*, and *follower*.

**Tweet.** In Twitter, both a post and posting action are called tweets. Twitter restricts the length of tweets to no more than 140-characters. Because of this restriction, people commonly use URL shortening services when they are posting URLs. Similarly, spammers use shortened URLs and few words to attract clicks.

**Following and follower.** Following someone means subscribing their tweets as a follower. If user A follows user B, B is following of A and A is a follower of B (see Fig. 1). The updates of user B automatically appear to user A. This is similar to Really Simple Syndication (RSS). Followings and followers are represented as edges in Twitter graph. A Following relation means out-edge and a follower relation means in-edge (see Fig. 1).

**Mention.** If @*username* is included in a tweet, it is called a mention. Mentions appear to a receiver even if the receiver is not a follower of the sender. It is almost the same as a message function on other social networking sites. Spammers commonly use this function to send spam because normal users rarely follow spammers. On Twitter, a reply is also considered a mention.

**Retweet** A retweet is a reposting another user's tweet. When a user finds a tweet that he wants to share with his followers, he can use the retweet function.

**Hashtag** The '#' symbol is a hashtag in Twitter. The hashtag is attached to the front of keywords to categorize tweets. This function is the same as a *tag* used in blogs. If a keyword is hashtagged a lot, it will appear in trending topics that appear to all Twitter users. Spammers often use trending topics in their tweets even though these topics are irrelevant to the contents of the spam messages. They also try to make trending topics using the keywords they want.

### 2.2 How Twitter Deals with Spam

Twitter users can report a spammer by clicking the "Report to @*username* for spam" menu on the spammer's profile page. Reported spammers are reviewed by the administrators and then suspended. Users can also report spammers by mentioning them to the official @*spam* account [17]. However, these manual methods require users' effort and there are many fake reports. Besides the users' reporting, Twitter has established several restrictions to prevent spam and abuse. The representative restrictions are as follows:

- Following a large number of users in a short time
- Following and unfollowing someone in a short time or repeatedly
- A small number of followers compared to the amount of following
- Multiple duplicated updates
- Updates mainly consisting of links

The above restrictions, however, are easy to avoid and spammers can always create new accounts even though their old accounts have been suspended. Still, about a hundred spam accounts are reported to the @*spam* account every day. Twitter published a blog post which stated that spam has been reduced as a result of their restrictions and that they constantly stand against spammers [18]. According to the posting, the percentage of spam per day has decreased from 11% in August 2009 to about 1.5% in February 2010. However, the data that only consists of percentages is difficult to analyze objectively. If legitimate tweets are increased much faster than spam, the percentage of spam is decreased. In fact, Twitter grew by about 1,400% in 2009 [19]. Moreover, there are about 140 million tweets per day [3]. This means that there may exist about a million spam messages, if 1% of tweets are spam.

## 3 Overview

We identify spam using the relation information between users. First, we measure the *distance* of user pairs. For example, when two users are directly connected by a single edge, the distance between the users is one. This means that the two users are friends. When some user pairs have a small distance longer than one, this means they have common friends although they are not friends themselves. In our experiment, almost all messages that come from a user whose distance is more than four are spam. Thus, the relationship is meaningless or untrustworthy when the distance is over four. If some user pairs have a distance greater than four, one of the users has very few relationships or no relationship like spammers. Therefore, we treat the messages coming from a user whose distance is greater than four as spam and we only identify the messages coming from a user whose distance is at least four.

The second feature is the *connectivity* between users. The connectivity represents the strength of the relationships. An edge may exist between a legitimate

user and a spammer when the spammer establishes a relationship with a legitimate user. Yu *et al.* called these edges *attack edges* [20, 21]. Each spammer has few attack edges because the spammers are difficult to establish relationships with legitimate users. Thus, the connectivity between a legitimate user and a spammer is weaker than the connectivity between legitimate users, when the distance is the same. We measure connectivity by using random walk and min-cut techniques. To evaluate our system, we collected a considerable amount of normal messages and spam messages from Twitter and identified the messages using their features. Distance and connectivity were not used in the previous work for detecting spam and they are difficult to be manipulated by the spammers. In addition, our system allows service managers or clients to identify each message in *real-time*. Thus, there is no delay, unlike in account-based methods.

### 3.1 Graph

To measure distance and connectivity, we used specialized subgraphs of the social network graph representing the relation between users. Twitter network can be represented by directed graph using following and follower relations. Our method focuses on the relation between the message sender and the receiver. Thus, we only construct the graph between them. Let a directed graph $G = (V, E)$ be an entire social network graph and $G' = (V', E')$ be a subgraph of $G$ satisfying the following conditions:

1. The graph $G' = (V', E')$ is a subgraph of a graph $G = (V, E)$.
2. The source node $s$ and terminal node $t$ are included in $V'$.
3. All nodes in $V'$ are included in the paths from $s$ to $t$.
4. All edges in $E'$ are included in the paths from $s$ to $t$.

We construct the graph $G'$ and measure the distance and connectivity between a node $s$ and a node $t$. In our case, the graph $G$ is the entire Twitter network graph, the node $s$ a message receiver, the node $t$ a message sender. Our system evaluates the sender on the receiver's position; thus, the paths from the receiver to the senders are considered. In the graph $G'$, all nodes are included in the paths from the receiver to the sender. There are three steps to construct the graph $G'$ of Twitter.

1. Put the receiver, his followings and followings of his followings to $V'$ and edges between them to $E'$.
2. Put the sender, his followers and followers of his followers to $V'$ and edges between them to $E'$. If the distance between the sender and the receiver is lower than four, $G'$ will be connected.
3. Remove the nodes which are not included in the paths from the receiver to the sender from $V'$ and edges to them from $E'$.

We only consider the paths whose length is at least four. Thus, we remove some nodes from $G'$ when they are only included in the paths longer than four. Fig. 2 shows a simple example of the graph. The reasons why we used the subgraph $G'$ are as follows:

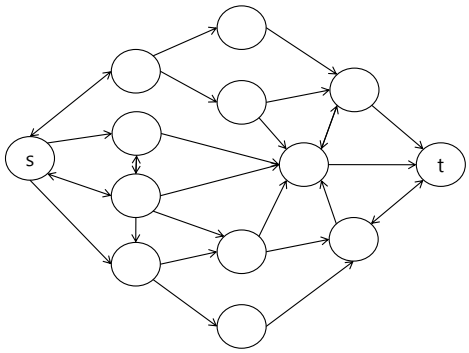



**Fig. 2.** A simple example of the graph when the distance is three.

- Analyzing the relation between the receiver and the sender is the most important task in this work. We do not need an entire network graph.
- The social network is huge. Twitter has about 190 million users. Thus, we cannot handle the whole social network.
- We use both the followings of the receiver and the followers of the sender to reduce crawling data. If we only use the receiver's followings, the amount of the crawling data will increase exponentially.
- We only analyze the user pairs whose distance is at least four. As noted above, the messages coming from a distance greater than four are mostly spam. Moreover, Kwak *et al.* showed that 70.5% of user pairs have paths whose length is four or shorter in the Twitter network [22]. Thus, our research covers most cases in Twitter.

### 3.2 Features

Spammers have different characteristics from non-spammers. Our design is based on an insight similar to the one used by Sybil series [20, 21]. In general, spammers are difficult to make relationships with non-spammers but they make a group with other spammers. Spam groups have only a few attack edges to honest regions. Thus, most non-spammers are not connected with spammers, or have long and weak connections. Based on these facts, we identify spammers using the distance and the connectivity between users.

**Distance.** We measure distance, which is the length of the shortest path between users. It is the same as the number of hops from a message receiver to a message sender. In Twitter, an out-edge is following, meaning the follower trusts the following. We examined the correlation between the distance and spammers. To investigate the distributions of spam and non-spam messages according to distance, we randomly selected 10,000 benign and an equal number of spam messages from our data set (see Fig. 3). Within a distance of two, only 0.9%

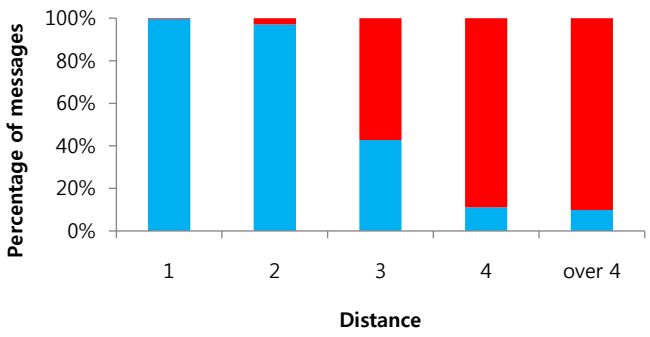



**Fig. 3.** The percentages of benign (blue) and spam messages (red).

messages are spam. However, 57.3% of the messages coming from a distance of three are spam and 89% of the messages coming from a distance of four are spam. From the result, most spam comes from users at a distance of more than three hops from receivers but there are also many benign messages at a distance of three or four hops. The connectivity feature discriminates between benign and spam messages that have arrived from the same distance.

**Connectivity.** The connectivity represents the strength of a connection. A simple way to measure connectivity would be counting the number of paths. More paths mean more friends are connected to the user. A better way to measure connectivity is counting the edge-independent paths. The collection of paths is called edge-independent if no two paths share an edge. We used Menger's theorem which characterizes that connectivity of a graph in terms of the number of independent paths between nodes [23, 24]. Menger's theorem defines edge-connectivity as follows:

**Theorem 1 (Menger's theorem).** *Let $G$ be a finite undirected graph and $u$ and $v$ be two distinct nodes. The size of the minimum edge cut for $u$ and $v$ is the same as the maximum number of the edge-independent paths from $u$ to $v$.*

This is a special case of the *Max-flow min-cut theorem*. The problem of finding the maximum number of the edge-independent paths can be transformed to a maxflow problem by constructing a directed graph assigning each edge with unit capacity. We compare the min-cut size when both nodes $s$ and $t$ are non-spammers, and when a node $s$ is a non-spammer and a node $t$ is a spammer. As expected, the min-cut sizes of the spammer's cases are smaller than that of the normal cases.

We also use random walk as another measure. Yu *et al.* used a special kind of random walk to identify sybil nodes, not exactly same as random walks [20, 21]. We used random walk technique used in PageRank [25]. The idea behind PageRank is that when a random surfer visits pages infinitely, the pages linked more are visited more. PageRank values are computed by the left eigenvectors $x_L$ of the transition probability matrix $P$ such that

$$x_L P = \lambda x_L,$$

where $\lambda$ is eigenvalue. The $N$ entries in the eigenvector $x_L$ are the steady-state probabilities of the random walk corresponding to the PageRank values of web pages. The *Perron-Frobenius Theorem* tell us that the largest eigenvalue of the matrix is equal to one which is the principal eigenvector [26, 27]. Thus, the principal eigenvector of the transition matrix $P$ is the PageRank values. We used this PageRank values. The web pages are corresponding to the users and the links are corresponding to the friendships. Because we use the specialized graph only including the nodes and edges in the paths from the node $s$ to the node $t$, the expected result of random walk is different from general graphs. All edges point toward the node $t$. Thus the eigenvector of the node $t$ is always top. Therefore, we convert the directed graph $G'$ to the undirected graph $G''$ replacing all directed to undirected edges. Now, both the nodes $t$ and $s$ have very high values in their eigenvector because the graph $G''$ is created by making backward-edges of existing edges. All random walks will proceed to both nodes $t$ and $s$ in normal cases. When the node $t$ is a spammer, however, the eigenvector of the node $t$ will not be as high as the node $s$ because the spammer only has a few edges.

## 4 Experiments and Evaluation

This section is composed of three parts. In the first part, we present how we collected data used in our experiments. In the second part, we show the spam detection results using the user relation features. In the last part, we show that the user relation feature can be represented as a user account feature to decide whether an account is a spam account or not. And we compare the results using only the account features used in the previous work and the results using the account features including the new one to detect spammers.

### 4.1 Data collection

Twitter offers API methods for data collection to encourage third-party developers, but there is a rate limit [28]. A host is permitted 150 requests per hour. Twitter also had a whitelist for developers but they stopped offering this whitelist on March 2011 [29]. In order to overcome the rate limit we used four servers and 120 IP addresses. The servers changed their IP addresses when they were stopped by the rate limit. The collection lasted for about two month from February to March 2011. We crawled 148,371 profiles, 267,551 tweets, 4,317,161 user's followings and 963,181 user's followers. We randomly selected non-spammers by using numerical Twitter user IDs. Spam accounts were selected from among the reported accounts to the "@*spam*" account, which is the official Twitter account. Legitimate Twitter users can report the spam accounts by mentioning to the "@*spam*" account; thus, we searched mentions using the "@*spam*" keyword and collected spam accounts from the search results. We manually checked whether each account is a spammer or not. In total, we collected 308 spam accounts and 10,000 spam messages.

**Table 1.** The results of classification using distance and random walk

| Classifiers | True Positive (%) | False Positive (%) |
|---|---|---|
| Bagging | 93.3 | 8.5 |
| LibSVM | 93.2 | 8.3 |
| FT | 93.1 | 7.7 |
| J48 | 92.3 | 8.7 |
| BayesNet | 92.0 | 8.0 |

### 4.2 Spam Classification

In the previous section, we proposed a spam filtering using user relation features. We identified spam using distance and connectivity features. Connectivity is measured in two ways: random walk and min-cut. First, we used the results of random walk with the distance. Given a graph $G''$, which is explained in Section 3, the result of random walk is the left eigenvector $x_L$ of the transition matrix of $G''$. Let $i$ be the index of a receiver and $j$ be the index of a sender in $x_L$. Then, their random walk values are $x_L[i]$ and $x_L[j]$, respectively. When the sender is a non-spammer, $x_L[i]$ and $x_L[j]$ are similar values and they are quite higher than the average value of $x_L$. When the sender is a spammer, however, $x_L[j]$ is much lower than $x_L[i]$. Therefore, we use the ratio $x_L[j]/x_L[i]$ as a feature from random walk. We randomly selected 5,000 messages where both senders and receivers are non-spammer, and 5,000 messages where senders are spammers and receivers are non-spammers from the data set. Then we constructed graphs for each user pair. On average, the graphs have about 5,000 nodes. We used Weka [30], which is a data mining tool, and used 10-fold cross validation option in classification . In $K$-fold cross validation, the sample data is randomly partitioned into $k$ subgroups. Only one partitioned data is used as validation data and the remaining $k-1$ partitioned data are used as training data. This process is then repeated $k$ times in order to use all $k$ subgroups as the validation data. Table 1 shows the results of applying each classifier. True positive means that spam messages are correctly classified as spam, which is 1 - false negative. False positive means that normal messages are classified as spam. All classifiers successfully identify spammers with about 92% true positive. Fig. 4 shows a decision tree created by the J48 classifier. The decision tree is simple, meaning that if the system uses the distance and random walk features, the system can easily identify the spammers.

Next, we selected 3,000 messages where both senders and receivers are non-spammer, and 3,000 messages where senders are non-spammer and receivers are spammer from the data set. The messages are classified using the results of min-cut and the distance. Finally, both results of random walk and min-cut were used with the distance in classifications at the same time. Table 2 and Table 3 show the results of the classifications. The classifiers also identify spammers with high accuracy when they only use the distance and min-cut results. In addition, the accuracy increases when the classifiers use the distance, the random walks and the min-cuts at the same time. From our experiments, we showed that we can identify spam using only relation information. This means

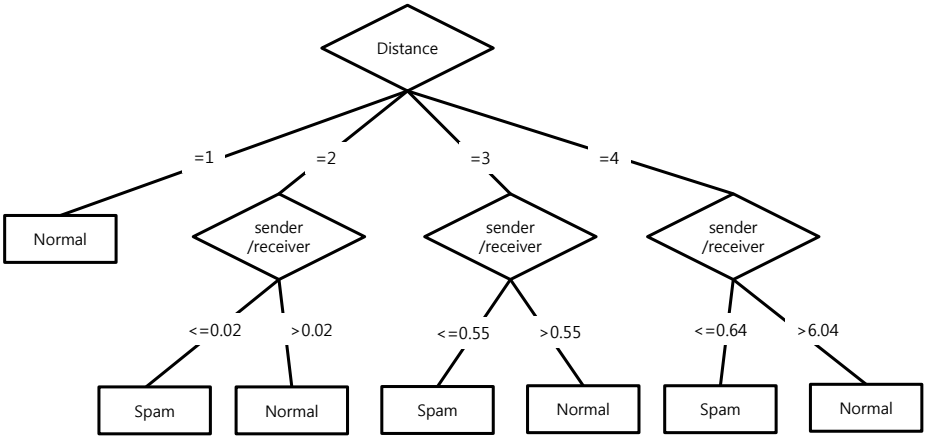



**Fig. 4.** A decision tree created by the J48 classifier

| Classifiers | True Positive (%) | False Positive (%) |
|---|---|---|
| Bagging | 94.6 | 6.5 |
| LibSVM | 94.0 | 5.8 |
| J48 | 93.9 | 5.3 |
| BayesNet | 93.5 | 5.5 |
| FT | 93.5 | 5.5 |

**Table 2.** The results of the classification using the distance and min-cut

| Classifiers | True Positive (%) | False Positive (%) |
|---|---|---|
| Bagging | 95.1 | 4.7 |
| LibSVM | 94.3 | 4.3 |
| J48 | 94.2 | 4.6 |
| FT | 93.8 | 4.4 |
| BayesNet | 93.4 | 5.9 |

**Table 3.** The results of the classification using the distance, random walk and min-cut

that our system can allow clients to decide whether or not received messages are spam in real-time. Fig. 5 shows Receiver Operating Characteristic (ROC) curves of classification results. When we use random walk and min-cut along with distance, the classification accuracy becomes better than when we use only distance.

### 4.3 Spam account detection with including a user relation feature

We consider that if we can include user relation related feature in the user account profile it would be easier to detect spam accounts.

One feature we consider is *the ratio of mentions sent to non-followers*. The distance of the messages sent to the followers is one. Non-spammers generally send messages to their followers or followings. On the other hand, spammers send messages to arbitrary users who are mostly located at a distance greater than one.

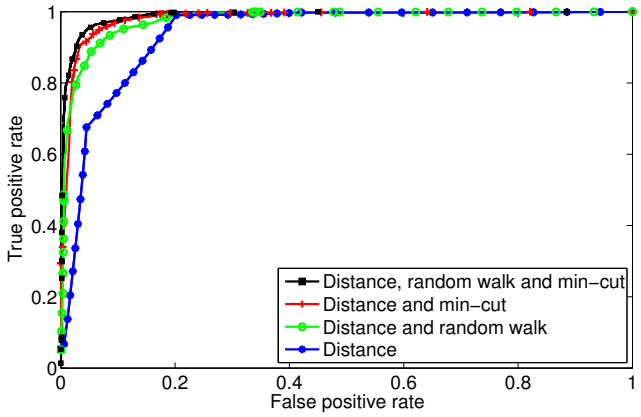



**Fig. 5.** ROC curves for each of the relation features.

We reproduced previous work's experiments related to detecting spam accounts in order to show that the results with adding our feature are better than those with only features used in previous work. The 11 features that are used in classifications are as follows:

- The standard deviation of tweeting interval
- The ratio of tweets containing URLs
- The ratio of mentions containing URLs
- The ratio of tweets containing hashtags
- The ratio of mentions ($\frac{|mentions|}{|total\ tweets|}$)
- The ratio of duplicate tweets
- Reputation ($\frac{|followings|}{|followers|}$)
- The number of lists including the user
- Age (the current time - the account creation time)
- The average content similarity
- The ratio of mentions sent to non-followers

*The ratio of mentions sent to non-followers* is the only relation feature and the others are account features which are used in previous work. The average content similarity is computed in the same as Lee *et al* [5]. They computed content similarity using the cosine similarity over the bag-of-words vector representation $\boldsymbol{V}(t)$ of the tweets:

$$similarity(t_1, t_2) = \frac{\boldsymbol{V}(t_1) \cdot \boldsymbol{V}(t_2)}{|\boldsymbol{V}(t_1)||\boldsymbol{V}(t_2)|}$$

Then, they measured the average content similarity over all pairs of tweets:

$$\sum_{t_1,t_2 \in \ set\ of\ pairs\ in\ tweets} \frac{similarity(t_1, t_2)}{|set\ of\ pairs\ in\ tweets|}$$

**Table 4.** The top five results of spammer detection using Weka classifiers

| Classifiers | True Positive (%) | False Positive (%) |
|---|---|---|
| BayesNet | 99.7 | 0.6 |
| LogitBoost | 99.7 | 0.6 |
| J48 | 99.6 | 0.6 |
| Logistic | 99.4 | 0.9 |
| LibSVM | 98.3 | 0.5 |

**Table 5.** The results of feature selection

| Rank | Information Gain |
|---|---|
| 1 | The ratio of mentions sent to non-followers |
| 2 | Reputation |
| 3 | The ratio of mentions containing URLs |
| 4 | The ratio of tweets containing URLs |
| 5 | Age |

| Rank | ReliefF |
|---|---|
| 1 | The ratio of mentions sent to non-followers |
| 2 | The ratio of tweets containing URLs |
| 3 | Age |
| 4 | The ratio of mentions containing URLs |
| 5 | The average content similarity |

| Rank | Chi Square |
|---|---|
| 1 | The ratio of mentions sent to non-followers |
| 2 | Reputation |
| 3 | The ratio of mentions containing URLs |
| 4 | The ratio of tweets containing URLs |
| 5 | Age |

We selected 1,000 non-spammers and 300 spammers from our data set and extracted the most recent 50 tweets from their timelines. The users were classified using several classifiers in Weka with a 10-fold cross validation option. Table 4 shows the top five results of classification among Weka classifiers. The accuracy is about 99.7% and the false positive is only about 0.6%. The accuracy are better than the spam classification in Section 4.2, but spam account detection methods cannot detect spam in real-time.

We also ranked the features to verify the importance. The feature selection methods used are also available on Weka, Information Gain, ReliefF and ChiSquare. Table 5 shows the five most important features for each method. All feature selection methods rank the ratio of mentions sent to non-followers as the top feature. It means that the relation feature is more powerful than the account features.

## 5 Discussion

### 5.1 Combination of account features and relation features

We only used relation features to detect spam in order to focus on the effect of the relation features. When a message is being delivered, our system verifies whether a sender is a spammer or not only using relation information between a message sender and a message receiver. The results are quite good but if we use both the account features and the relation features, the spam filtering system will be more powerful. In Section 4.3, we used both the account features and the relation feature. The accuracy is better than the results when only used the relation features. The account features supplement the relation features' insufficiency.

### 5.2 Live detection

Our system can be applied to both client-side and server-side. When our system is applied to client-side, the system should collect relation information periodically from Twitter. The distance and the connectivity are computed using collected data. In these processes, the client needs some bandwidth, computing, storages resources and time. Most of received messages, however, come from the client's friends. The messages coming from the friends do not need to identify senders. Therefore, there will be only a few cases that crawling the data and computing relation features for indentifying the sender. Given those facts, the resource problems are not big. When our system is applied to server-side, it is more practical. Additional bandwidth and storage resources are not needed because service managers already have user's relation information. However, the service managers should compute all users' relation features. It may cause a heavy load to the server, so they should prepare separate computing servers. Computed relation features will be cached and then only updated when the relation features are changed. Caching technique will help both client-side and server-side to reduce computing overhead.

### 5.3 Limitations

Spammers have very few relationships or no relationships with normal users. This is the reason why our system checks the message sender by computing the distance and the connectivity from the message receiver to the message sender. However, this method has two problems. First, if a normal user creates a new account and sends a message to his friend before the new account has any followers, the message will be filtered. This is because new account's characteristics are same as spammer when the new account is created and it has not established any relationships yet. This, however, is a temporal problem because the new account will get followers soon. The second problem is that our system will identify the messages as normal even though the messages come from infected friends. Sometimes attackers send spam through normal users' accounts

by using Cross-Site Request Forgery (CSRF) or password stealing. Also, many malicious applications use crafty tricks for getting a writing permission of normal users. The innocent and careless users allow that the applications can write postings using the user's own name. Infected users' friends receive spam from his infected friends. Only checking relation features cannot solve this problem. When a user sends the messages using the application that has never been used by the user, the messages should be suspected. Ultimately, the contents of the messages should be checked whether the contents are spam or not. Because of tweet's short length, identifying only the URLs contained in the messages is a good solution. There are related work about classifying web pages into spam or not [31–33].

## 6 Conclusion

In social networks, traditional spam filtering methods are not effective because of the characteristics of social networks. We propose a spam filtering method for social networks using relation information between users. We use distance and connectivity as the features which are hard to manipulate by spammers and effective to classify spammers. Moreover, our system identifies spam in real-time because it does not need a user history data. Services managers or clients can decide whether or not the messages are spam. We hope that our system contributes to quarantine a suspected message into spam message box in social networking services. Also, we showed that user relation concept can be reflected into user account profile to detect spam accounts. We evaluated the system using Twitter data but the system is also effective for other social networking services because all such services contain relation features.

## Acknowledgement

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)" (NIPA-2011-C1090-1131-0009) and WCU(World Class University) program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (R31-2010-000-10100-0)

## References

1. Alexa: Top sites in united states (2011) `http://www.alexa.com/topsites/countries/US`.
2. NielsenWire: Twitter's tweet smell of success (2009) `http://blog.nielsen.com/nielsenwire/online_mobile/twitters-tweet-smell-of-success/`.
3. TwitterBlog: #numbers (2011) `http://blog.twitter.com/2011/03/numbers.html`.

4. Webb, S., Caverlee, J., Pu, C.: Social honeypots: Making friends with a spammer near you. In: Proceedigns of the Fifth Conference on Email and Anti-Spam (CEAS). (2008)
5. Lee, K., Caverlee, J., Webb, S.: Uncovering social spammers: Social honeypots + machine learning. In: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retreival, ACM (2010)
6. Stringhini, G., Kruegel, C., Vigna, G.: Detecting spammers on social networks. In: Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC), ACM (2010)
7. Markines, B., Cattuto, C., Menczer, F.: Social spam detection. In: Proceedings of the 5th international workshop on Adversarial Information Retrieval on the Web (AIRWeb), ACM (2009)
8. Yardi, S., Romero, D.: Detecting spam in a twitter network. First Monday **15**(1) (2010)
9. Gayo-Avello, D., Brenes, D.J.: Overcoming spammers in twitter - a tale of five algorithms. In: 1st Spanish Conference on Information Retrieval. (2010)
10. Wang, A.H.: Don't follow me: Spam detection in twitter. In: Proceedings of 5th International Conference on Security and Cryptography (SECRYPT). (2010) 142–151
11. Benevenuto, F., Magno, G., Rodrigues, T., Almeida, V.: Detecting spammers on twitter. In: Proceedings of the 7th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS). (2010)
12. Chu, Z., Gianvecchio, S., Wang, H., Jajodia, S.: Who is tweeting on twitter: Human, bot, or cyborg? In: Proceedings of Annual Computer Security Applications Conference (ACSAC). (2010)
13. Thomas, K., Grier, C., Ma, J., Paxson, V., Song, D.: Design and evaluation of a real-time url spam filtering service. In: Proceedings of the IEEE Symposium on Security and Privacy. (2011)
14. Blanzieri, E., Bryl, A.: A survey of learning-based techniques of email spam filtering. Artif. Intell. Rev. **29** (March 2008) 63–92
15. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A bayseian approach to filtering junk e-mail. In: Learning for Text Categorization: Papers from the 1998 Workshop. AAAI Technical Report, AAAI Technical Report WS-98-05 (1998)
16. Grier, C., Thomas, K., Paxson, V., Zhang, M.: @spam: the underground on 140 characters or less. In: Proceedings of the 17th ACM conference on Computer and communications security, ACM (2010)
17. TwitterHelpCenter: How to report spam on twitter `http://support.twitter.com/articles/64986-how-to-report-spam-on-twitter`.
18. TwitterBlog: State of twitter spam (2010) `http://blog.twitter.com/2010/03/state-of-twitter-spam.html`.
19. TwitterBlog: Measuring tweets (2010) `http://blog.twitter.com/2010/02/measuring-tweets.html`.
20. Yu, H., Kaminsky, M., Gibbons, P.B., Flaxman, A.: Sybilguard: defending against sybil attacks via social networks. In: Proceedings of ACM SIGCOMM Conference. SIGCOMM '06, ACM (2006)
21. Yu, H., Gibbons, P.B., Kaminsky, M., Xiao, F.: Sybillimit: A near-optimal social network defense against sybil attacks. In: Proceedings of the 2008 IEEE Symposium on Security and Privacy, IEEE Computer Society (2008)
22. Kwak, H., Lee, C., Park, H., Moon, S.: What is twitter, a social network or a news media? In: WWW '10: Proceedings of the 19th international conference on World Wide Web, ACM (2010)

23. Menger, K.: Zur allgemeinen kurventheorie. Inventiones Mathematicae **10** (1927)
24. Aharoni, R., Berger, E.: Menger's theorem for infinite graphs. Inventiones Mathematicae **176**(1) (2009)
25. Langville, A.N., Meyer, C.D.: A survey of eigenvector methods for web information retrieval. SIAM Review **47**(1) (2005)
26. Perron, O.: Zur theorie der matrices. Mathematicsche Annalen **64**(2) (1907)
27. Keener, J.: The perron–frobenius theorem and the ranking of football teams. SIAM Review **35**(1) (1993)
28. TwitterAPIwiki: Rate limiting `http://dev.twitter.com/pages/rate-limiting`.
29. Paul, R.: Twitter tells third-party devs to stop making twitter client apps (2011) `http://arstechnica.com/software/news/2011/03/twitter-tells-third-party-devs-to-stop-making-twitter-client-apps.ars`.
30. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., H., I.: The weka data mining software: An update. SIGKDD Explorations **11**(1) (2009) `http://www.cs.waikato.ac.nz/ml/weka/`.
31. Ntoulas, A., Najork, M., Manasse, M., Fetterly, D.: Detecting spam web pages through content analysis. In: Proceedings of the 15th international conference on World Wide Web. WWW '06, ACM (2006)
32. Thomas, K., Grier, C., Ma, J., Paxson, V., Song, D.: Design and evaluation of a real-time url spam filtering service. In: Proceedings of the IEEE Symposium on Security and Privacy. (May 2011)
33. Ma, J., Saul, L.K., Savage, S., Voelker, G.M.: Identifying suspicious urls: an application of large-scale online learning. In: Proceedings of the 26th Annual International Conference on Machine Learning. ICML '09 (2009)