2014

Discovering Spam On Twitter

Ioana-Alexandra Bara
Virginia Commonwealth University

DISCOVERING SPAM ON TWITTER

A Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at Virginia Commonwealth University.

by

IOANA-ALEXANDRA BARA

August 2013 to August 2014

Director: Dr. Carol Fung,

Dr. Krzysztof Cios, Department of Computer Science

Virginia Commonwewalth University

Richmond, Virginia

August, 2014

Acknowledgements

I would like to sincerely express my gratitude to Dr. Carol Fung for her guidance throughout my Masters. Dr. Fung has shared her knowledge with me and made sure I am on track and that I keep working towards bettering myself.

Along with Dr. Fung, I would also like to acknowledge Dr. Than Dinh, for his most helpful sharing of knowledge and guidance. Dr. Dinh has been a key tool towards developing my algorithm and I sincerely appreciate all the help I received from him.

TABLE OF CONTENTS

Chapter Page

List of Algorithms

LIST OF TABLES

LIST OF FIGURES

Abstract

DISCOVERING SPAM ON TWITTER

by Ioana-Alexandra Bara

A Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at Virginia Commonwealth University

Virginia Commonwealth University, 2014.

Director: Dr. Carol Fung,

Dr. Krzysztof Cios, Department of Computer Science

Twitter generates the majority of its revenue from advertising. Third parties pay to have their products advertised on Twitter through: tweets, accounts and trends. However, spammers can use Sybil accounts (fake accounts) [21] to advertise and avoid paying for it. Sybil accounts are highly active on Twitter performing advertising campaigns to serve their clients [5]. They aggressively try to reach a large audience to maximize their influence. These accounts have similar behavior if controlled by the same master. Most of their spam tweets include a shortened URL to trick users into clicking on it. Also, since they share resources with each other, they tend to tweet similar trending topics to attract a larger audience. However, some Sybil accounts do not spam aggressively to avoid being detected [22], rendering it difficult for traditional spam detectors to be effective in detecting low spamming Sybil accounts. In this paper, I investigate additional criteria to measure the similarity between accounts on Twitter. I propose an algorithm to define the correlation among accounts by investigating their tweeting habits and content. Given known labeled accounts by spam detectors, this approach can detect hidden

accounts that are closely related to labeled accounts but are not detected by traditional spam detection approaches.

CHAPTER 1

INTRODUCTION

Social Networks have become a very popular mean for people to communicate. Today's most popular social platforms are Facebook, Twitter, Tumblr, Instagram and others. People use these platforms to communicate through messages, posts, tweets etc., share pictures and generally keep in touch with friends or follow their favorite celebrities. Companies and businesses use them to advertise their products and services, while governmental institutions use them to inform/educate consumers and for emergency situations [3].

The first time an online article covered a story about Twitter was in 2006. At that time, the service that was later call Twitter, was supposed to be a group send SMS application where each person could send a message to its own network of friends [2]. Twitter started as a project in a small company named Odea, founded by a few ex-Googlers [10]. It was initially a start-up among friends. The company quickly evolved into a multi-billion dollar star-up. Today, the majority of the revenue Twitter makes is from advertisement. Advertisement on Twitter has increased by 119% over the last from June 2012 to June 2013 which brought the company a roughly 113 million in revenue increase. Advertisement reaches an incredible number of users on Twitter and the revenue it produces represents 85% of Twitters overall revenue [17]. Fake or compromised accounts use advertising as means for spamming or phishing. The latter one can be very dangerous to legit users, since acquiring sensitive information such as user-names and passwords can be potentially used to access credit card information. Not all advertisement on Twitter is spam, many companies pay Twitter to be able to endorse their products or services. However, not all companies can afford to pay for online advertisement, or are willing to do so. This is the reason why some of them would appeal to third parties to buy fake accounts

and promote product and web pages versus paying Twitter.

These days you can buy as many bot accounts as you like, that will behave in such a manner to aggressively advertise. According to this article [13] $11 dollar will buy you on average 1000 fake accounts. Fake account sellers can be found on eBay(52), 55 different follower selling websites can be found on Google, while 6400+ Twitter followers services are found on Fiverr.com .

Twitter has means to protect its users from spam. They have a spam detection algorithm that will flag tweets as potentially harmful. The algorithm focuses on criteria such as harmful links, aggressive following behavior, posting repeatedly to trending topics, posting duplicated tweets, posting links with unrelated tweets etc. [15].The algorithm allows the social network platform to discover and terminate accounts. There is a constant need in evolution of the way spam accounts are discovered. According to this article [11] an underground fake account seller is able to change the behavior of its fake accounts to avoid detection by filling out more account details (pictures, personal information) and create some activity on the accounts before selling them. If Twitters flagging system was able to detect fake accounts with incomplete profiles (no pictures and little activity), it has now become incapable to detect these new accounts that simulate real accounts. From re-tweeting each others tweets, to making small text alteration of the tweeted text, malicious accounts find a way to avoid detection.

My approach to discovering spam on Twitter is different than previous approaches, in that it does not focuses on the account details (pictures, number of friends and followers etc.), it emphasis on the interaction between accounts, the way they tweet and re-tweets, the patterns they use and the number of tweets each account has, in particular the number of spam tweets. My contribution consists of means to identify an entire web of spam accounts that have a close relationship by posting similar tweets and exercising similar behavior within a community.

The research shows that fake account holders have found means of avoiding detection, by simple adjustments made to the account information. It is no longer efficient to only analyze the

account structure, but it is now necessary to analyze the interaction among accounts as well as the tweeting pattern they exercise.

The layout of the thesis consists of a Literature Review Chapter, where I present previous research done in the area of Twitter spam. Following is the Problem Description chapter that presents the need of a spam detection evolution on Twitter and the algorithm devised for spam detection. The Data Collection chapter will then introduce the methods used to collect data from Twitter as well as an initial analysis of the data. Chapter 5, Methodology, will introduce the two methods used to discover new spam followed by the Results chapter that will give a detailed explanation of the spam that was discovered. The final chapter will be Conclusion and Future Work, with an overview of the detection mechanism and results.

CHAPTER 2

LITERATURE REVIEW

Social Networks have revolutionized the way people interact online. With online platforms for anything from dating(i.e. Match.com) to selling product online(i.e. Amazon, eBay) and education(i.e. ITT Technical Institute) everything is withing a mouse click. Companies use these platform to endorse their products and increase consumer reach.

2.1 Social Network Advertisment

Online social networks, also know as web-based services were introduced in the early 2000s. MySpace and LinkedIn were introduce in 2003, Flickr, Youtube, Facebook and Twitter to follow. The authors of [12] summarize that advertisements, programs and subscription-based services remain the main source of revenues for networking websites. The larger the number of users of the online platform, the more revenue the platform generates. According to a May 2013 UPS study on Customer Experience conducted by comScore [7], the US Retail e-Commerce Sales Growth has increase by 15% from 2011 to 2012 reaching an all time high of $186 Billion. The shoppers survey revealed that 84% indicated using at least one social media site. Facebook ranks as the most popular by a considerable margin, followed by Twitter, LinkedIn and Pinterest.

The growth of online shoppers changes the way companies advertise. While some percentage of this advertisement is paid for by companies, the majority of the advertisement is conducted through the use of free tools. Business Insider reports that the majority of advertisers(89%) and advertisement agencies(75%) use free tools [6].

2.2 Spam Bots and Sybil Accounts

There are ways to take advantage of free online advertisement and many agencies and companies rely on Spam Bots or Sybil accounts. These fake accounts pretend to be legitimate-distinct users and their behavior is similar [23]. Some of these accounts might seem surprisingly akin to being legitimate, payed for advertisement accounts. They cause the social network platform millions of dollars in revenue loss each year [17]. While there already exists different methods to protect against Sybil accounts, the authors of [1] propose new and more accurate means of detection. They propose a community detection algorithm that is based on random walks and white-listing honest nodes ranked by their trustworthiness. They rely on the assumptions that Sybil accounts have knowledge on the underlying network structure. They propose that Sybil defenders build local defense systems for each community, rather than rely on a global defense system. They argue that such discovery methods that imply random walks in local communities are more accurate at discovering Sybil accounts. Based on the same assumption the authors of [8] illustrate a methodology of labeling nodes in the network as honest user or Sybils.

2.3 Twitter Spam

Discovering Spam Campaigns on Twitter can dilute an entire community of spamming accounts that harasses honest Twitter users. Previous research has been made in the area of Twitter Spam, that detects unique spamming accounts given different criteria. For example dormancy (activation time since the account creation), abusing shortening services for URL's, abusing free hosting and sub-domains to host their content, and using URL domains with high reputation. Individual accounts identified as spam are suspended by Twitter. A spam account can be deleted by the social platform, with the use of its own spam detection methods. Research done in the area of social media spam proposes different criteria to identify malicious

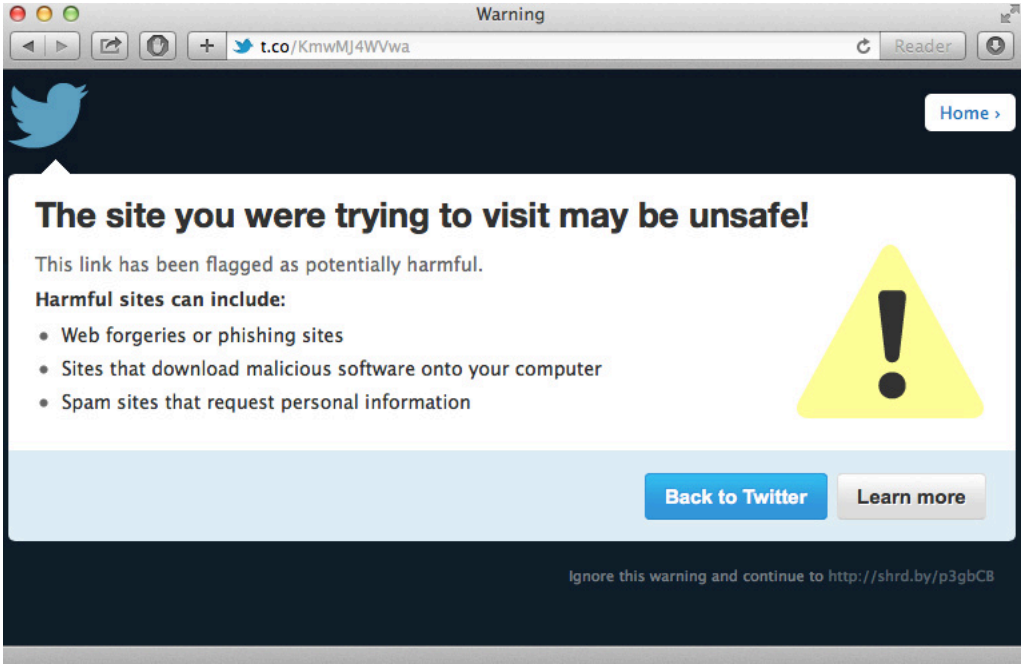

Fig. 1.: Twitter Warning Web page

accounts. For example, the author of [19] proposed criteria the involves looking at embedded links that lead to malicious sites, posting duplicated tweets, sending unsolicited replies and mentions and using trending topics. This was the first paper to propose automated means to detect spam. Shridharan et. al. analyses means by which spam accounts pick their targets with the purpose of reaching as many accounts as possible. They analyze the gathered data to discover which methods spam accounts use and how they are used. All these different criteria to detect spam focuses on individual accounts and does not take in consideration the relationship between accounts. I propose to focus on a community of accounts that act similar, in order to detect not one, but an entire web of accounts that form spam campaigns. The initial analysis of the collected data, detailed in section 4.4, allows to select some of the features of the previously mentioned spam detection mechanisms and at the same time introduce new criteria to formulate an efficient spam detection algorithm. In [19] the author uses the ratio of number of friends to followers and duplicate tweets along with other criteria, to decide whether the

account is spam. However, the initial analysis shows that this criteria will not hold against current evolved spammers. Another observation based on the initial analysis shows that we cannot rely on the fact that spam accounts tend to tweet duplicate tweets, as mentioned in [19]. Spam has evolved by altering a few characters in tweets that makes duplicated tweet detection impossible. Further elaboration on this topic in section 4.4. The same author used another technique of removing mentions of other users, links and hash tags from a tweet. The purpose of this is to try to match exact tweets. However, I discovered that this is not enough to get an exact match. Tweets need to be stripped of all non-characters, digits and white spaces in order to get a closer match. In [14] the authors make use of a word frequency count by analyzing spam tweets and then query for more tweets that contain these frequent words. I applied this technique to the collected data, to see if the results are conclusive with the expectations. However, the experiment resulted in a large number of false positives, because spammers typically avoid using frequent words, unless they make use of trending hash tags. Analyzing the data set by stripping the tweets of stop words (most common used words in the English vocabulary such as 'however', 'again', 'meanwhile' etc) resulted in being unable to conclude whether a certain account could potentially be spam or not due to too many overlaps with trustworthy users. Previous research in the field [5] uses a similar model like the one I propose in Chapter 5 of assigning a spam score to accounts on Online Social Networks. The research however is not specifically applied to Twitter and uses random walks to discover fake accounts.

For this research, I am looking to identify machine generated accounts that exercise similar behavior, and are under the control of the same master. I am aware that, there exist spam accounts that are maintained by humans [16], and act in a different manner than machine generated accounts. The difference lies in the way the two accounts tweet and maintain their profile. As mentioned in [20] Crowdsourcing Sybil detection is very efficient in discovering spam accounts when the participants pay close attention to all the accounts they label. However, this way of detection is only applicable on a small scale, as it is time and resource consuming and

hence needs to be automated. According to [9] 67% of internet users use Social Networking Site and 16% use Twitter. Manual discovery of spam accounts is inapplicable on such a large scale. While machine maintained spam accounts might have a few random pictures, or simply artistic pictures of nature or sports etc., maintained accounts can have multiple pictures of a single person, possibly videos and all tweets are very different in wording. Such accounts would not build a tweeting pattern are very hard to distinguish from real accounts. I consider such accounts as being aggressive advertisement and am not specifically looking for them in the data set.

2.4 Difference between my method and other methods

The spam detection method I introduce, makes use of some of the techniques used in previous research papers, that were mentioned in the overhead section. For instance, I look at tweeting patterns, crawl the URLs embedded in tweets and also analyze the frequency of repeated tweets. More on the details of these methods in Chapter 5. I introduce new methodologies such as comparing tweeting patterns among honest and dishonest users, labeling users as dishonest based on similar tweets with spam accounts and assign spam scores to accounts based on the interaction among them. This is a novel approach to discovering spam and the process is explained in detail in Chapter 3.

CHAPTER 3

PROBLEM DESCRIPTION

There are multiple ways one can get exposed to spam on Twitter. The first one is through a users' timeline, where tweets or re-tweets are displayed. One can also get exposed to spam by looking at trending topics. A direct message is also popular among spammers, since on Twitter anyone can receive a message regardless of the connection between the two accounts. Spammers also embed popular search terms within their tweets, because a user can search any tweets and trending topic by using the Search tab. According to previous research, 68.3% of successful spam [3] use regular tweets. Hence I focus on regular tweets to analyze the likelihood of an account to be labeled as spam. Previous research also shows that spam accounts use URLs embedded in their regular tweets along with popular hash tags. For this reason, I focus on the investigation of the URLs used in tweets. Since Twitter uses URL shortening helpers, users of Twitter have no means of telling what domain names the URL links to. Twitter has an algorithm for detecting malicious links and uses it to warn users. When a spam flagged link is clicked on by the user, Twitter will post a warning and allow the user to follow the malicious link or return to Twitter (see Fig 1). My purpose is to detect accounts that post such potential harmful links, that have been marked by Twitter and use them to discover new spam. By analyzing accounts that have already tweeted spam tweets, I propose an algorithm to discover malicious accounts that initially might seem harmless. The algorithm makes use of the connection among different accounts, given their tweeting patterns.

Spam on online social media is considered aggressive advertisement. An example of aggressive advertisement is unsolicited mentions that encourages a user to click on a link. Not all advertisement on online social media is spam, but the focus of my research is to discover ad-

vertisement that Twitter has not been paid for.

Twitter would flag a tweet if it has been reported as spam by a number of users. Most of these tweets are tweeted by harmful accounts that are either maintained by a human or posted by machine generated accounts. There is a fine line in deciding whether an account is handled by a human or a computer, some trends can be observed for the computer generated accounts or tweets:

- Tweets tend to have a similar tweeting pattern.
- The accounts do not contain any pictures, or if they do contain pictures they can usually be found on Google Images and the images tend to be unrelated to each other.

It is generally a hard problem for a computer to come up with original tweets, that are grammatically correct. For this reason, spam accounts would reuse tweets, where they alter a digit, a character, add or remove a mention etc. I focus solely on machine generated accounts that make use of tweeting patterns.

The algorithm I am going to present next uses tweeting patterns and initially flagged tweets to discover new spam. The purpose of the algorithm is to assigns a final spam value to each tweet and account. This spam value represents the likelihood of the tweet or account to be spam. The following algorithm presents the steps necessary to discover malicious accounts and tweets:

Algorithm 1 Discovering Spam Accounts

Require:

- Set $U := \{u_1, u_2, ...u_n\}$ set of all tweets that contain a link
- Set $X := \{x_1, x_2...x_m\}$ set of all users that have posted a tweet in U

for each unchecked $u_j \in U$ **do**
  if u_j was flagged **then**
    set initial spam for u_j^0=1
  else
    set initial spam for u_j^0=0
  end if
end for
for each $x_i \in X$ **do**
  set initial spam of x_i=0
end for
for all distinct tweets that have an initial spam value=1 **do**
  set the initial spam value of all other tweets in the database with the exact same pattern equal 1.
end for
run Spam Likelihood Algorithm
for each $u_j \in U$ **do**
  if u_j spam score > 0.1 **then**
    mark u_j as spam
  end if
end for
for each $x_i \in X$ **do**
  if x_i spam score > 0.1 **then**
    mark x_i as spam
  end if
end for

The algorithm starts by assigning an initial spam value to each tweet and account. Initially all accounts are viewed as honest and their initial spam value is set to zero. For tweets the initial spam values can be either 1 or 0. A tweet receives an initial spam value of 1 if the embedded link leads to a warning site. That warning site will prompt the user that the link might be malicious and gives the option to proceed to the website or return. From the initial analysis of the database I was able to conclude, that although some tweets are an exact match, some might be flagged as spam, while other are not. For this reason, I added the pattern matching technique(described in detail in section 5.1.1) to the spam detection algorithm, in order to discover those hidden spam tweets. This ensures that the tweets that have the same link embedded or the same tweeting

pattern and are initially flagged, all get an initial spam value of 1. The next step is the spam likelihood detection. This itself is another algorithm that assigns a spam likelihood value to each tweet and user. This algorithm analyzes interaction between users and use of similar tweets. When a user tweets multiple spam tweets, the user will also be assigned a high spam likelihood score. The same idea applies to tweets. If a tweet is tweeted by many spam accounts, the tweet will receive a high spam likelihood score. More on this algorithm is presented in chapter 5. When this step terminates each tweet and user will have an assigned spam likelihood score. The next step is to analyze this value and see if it is above a set threshold. If the spam value of the tweet or account is higher than the threshold, I conclude that there is a very high likelihood for this tweet or account to be malicious. This chapter introduces the algorithm used to discover spam. The following chapter will present the data collection process as well as an initial analysis on the collected data.

CHAPTER 4

DATA COLLECTION

4.1 Pre-requisites

Collecting new data from Twitter was necessary because all the available data to download was outdated. Spam evolves in order to avoid detection and for this reason new data was necessary to develop a detection mechanism that is able to capture all spam. In order to collect this data, I first had to register an application with Twitter developers. The following credentials are needed in order to have access to the data:

- Consumer Key
- Consumer Key Secret
- Access Token
- Access Token Secret

Apart from these tokens, the applications needs to be registered through a Twitter account that requires a user name and password. Once all these requirements are met, Twitter will allow the application to access the API and start streaming data. The Twitter API offers a Java interface that allows a registered application to download user information, tweets, messages and other. I collected data over the course of 6 weeks. There is a time limit constrain for data collection(see code in Appendix A.), the application was bounded by 180 user information per 15 minutes and 200 tweets per user. At the end of the 4 weeks there were approximately 10 Million tweets and 51.000 user information collected. For each user I registered the number of followers and friends. I used the Java API provided by Twitter for the data collection and stored them in a

mySQL database. Below is a snapshot of the Java code used to collect the user-name, friends and followers count and tweets given a users id which was previously saved in the database. The rest of the code from this source file can be found in Appendix A.

```
String screenName,text,date;
int follCount,friendsCount;
boolean isRetweet=false,isRetweeted=false;
User user;
try{
    for(int i=0;i<51000;i++){
        currentID = dbId.get(i);
        Paging paging = new Paging(1, 200);
        String name = db.getName(currentID);
        //ResponseList<Status> list1 = twitter.getMentions();
        user=twitter.showUser(name);
        boolean isProtected = twitter.showUser(name).isProtected();
        if(isProtected)
            System.out.println( user.isProtected()+"\n" + "protected user" + "\n");
            else{
                System.out.println("get status return is: "+user.getStatus());
                ResponseList<Status> list2 = twitter.getUserTimeline(name, paging);
                for (Status tweets : list2) {
                    date = ""+tweets.getCreatedAt();
                    text = tweets.getText();
                    screenName = tweets.getUser().getScreenName();
                    follCount = tweets.getUser().getFollowersCount();
                    friendsCount = tweets.getUser().getFriendsCount();
                    isRetweet = tweets.isRetweet();
                    isRetweeted = tweets.isRetweeted();
                    if(isRetweet)
                        db.addTweet(currentID,text,2,date);
                    else if(isRetweeted)
                            db.addTweet(currentID,text,3,date);
                    else db.addTweet(currentID,text,1,date);
                                }
            }
    }
}catch(Exception e){e.printStackTrace();};
```

Fig. 2.: Java code snapshot for data collection

4.2 Data Collection Process

The data collection algorithm started with a random Twitter account from which I further downloaded a list of followers. I randomly selected another 200 followers from that list. From there on more friends and followers were selected to download. Previous research showed that Amazon and Toyota were running major advertisement campaigns on Twitter towards the end of 2013. I also downloaded random users from the list of followers of each of these companies.

The data collection proceeded with 200 tweets per each user(limit imposed by Twitter).

4.3 Data Storage

The information collected was stored in mySQL. For each user I kept track of the count of friends and followers, the unique Twitter ID (composed of entirely digits) and the user-name. Further more for each unique ID, at most 200 tweets were downloaded. The following table represents a snapshot of the database :

Table 1.: User Information

ID	user-name	Followers	Following
***73	**omaslomez	253973	joewilcox
***5082	**nca	291	335
***33002	**iftmed	4960	60
***63033	**sanbeebe	11157	7868
***74229	**uhnke	9	123
***91079	**fXmal	42	218
***82278	**rsti	5026	1390

4.4 Initial Analysis

This section describes the particular patterns observed through a preliminary analysis. I was able to conclude, that tweets belonging to the same spam account display a similar pattern. The following observations could be made about tweets belonging to spam accounts:

1. Tweets can be an exact match.
2. Similar use of alpha-numeric characters but different embedded URL
3. Similar use of alpha-numeric characters but different user mentions

4. Similar use of alphabetic characters but different use of digits

Exact match of a tweet means different users have tweeted the same exact tweet, with the same embedded URL. For instance I observed more than 10 different users tweeting the following tweet at different times, during the same month of the same year:

- Win a new Macbook Pro on Cyber Monday 2009. Details here: http://bit.ly/29rFES

Similar use of alpha-numeric characters with a different embedded URL can be observed in the following example:

- Quick way to burn off 2+ inches from your waist while losing up to 30 lbs of body fat in less than 25 days http://t.co/EVlYCdXoFk
- Quick way to burn off 2+ inches from your waist while losing up to 30 lbs of body fat in less than 25 days http://t.co/YxrK6FrRti
- Quick way to burn off 2+ inches from your waist while losing up to 30 lbs of body fat in less than 25 days http://t.co/eChjLFtq1P

The above tweets have been tweeted by different users at different times, and they all make use of a different embedded URL. Spam accounts are constantly trying to avoid being detected, which is why their behavior evolved over time. For example, some accounts would use mentions of other users included in tweets:

- @***n_Ivan_Ivan Faucette http://t.co/YJeZ9Bd6
- @***eviva Buehler http://t.co/YJeZ9Bd6
- @***ovMark Wells Michigan MI http://t.co/XPmqaTs0

The following tweets illustrate an example of spammers changing a few digits to try and evade automatic spam detection:

- My Twitter account is worth $282.00, according to SocialTracker. See how much you are worth: http://t.co/oj9H9z174i
- My Twitter account is worth $88.80, according to SocialTracker. See how much you are worth: http://t.co/Rne3YBtw45

An snapshot of the spam database illustrating similar tweets can be seen in the following table: Another behavioral evolution of spam tweets is re-tweeting random tweets, to add diversity to

Table 2.: Spam Tweets

Index	shortURL	Tweet	User ID
530	http://t.co/m475afL	Hey Hey I just made $730 today working a few hour from home ! http://t.co/m475afL	***0712
531	http://t.co/1NKethu	I made $415 today working a few hour from home http://t.co/1NKethu	***0712
532	http://t.co/nA5K2Ok	Hey everyone I made $488 today working a few hour from home ! http://t.co/nA5K2Ok	***0712
446	http://t.co/mvIHrig	hey everyone youve gotta check out this article! I made $350 today so far http://t.co/mvIHrig	***1692
447	http://t.co/M0Nk7uB	check this out! I made $310 today so far http://t.co/M0Nk7uB	***1692
225	http://t.co/5x3IGeaE	Lose 20 lbs. of Fat in 30 Days - Without Doing Any Exercise http://t.co/5x3IGeaE	***469486
232	http://t.co/CBQJyHys	Fastest way to lose weight while removing toxins & boost energy levels http://t.co/CBQJyHys	***469486

their accounts. By manual check of some possible spam accounts, I can concluded that such re-tweeting of random tweet happens on average once every 20 regular tweets. The following pictures illustrates a snapshot of a spam accounts time-line:

Fig. 3.: Spam Account#1 use of re-tweet

Table 1 shows and example of the number of friends and followers of randomly picked unique user ID's from the database. A more in-depth analysis of all the spam labeled accounts results in the average number of 682 friends and 1088 followers per account. I observed that there is no pattern of the number of friends or followers of spam accounts. While some might have a limited number (below 100), some obvious spam accounts were followed by 10000+ friends. This chapter gave an overview of the collected data and the processes involved in

Table 3.: Friends and Followers Count

ID	Followers	Following
id1	52	2675
id2	145	208
id3	12089	223
id4	5	16
id5	356	757
id6	24	83
id7	299	373

Fig. 4.: spam Account#2 use of re-tweet

the data collection as well as the programming languages used. An initial analysis of the data resulted in a few observations (such as the use of tweeting pattern) that will be used in the Spam Discovery Algorithm presented in the upcoming chapters. The next chapter will present in detail the different methodologies formulated to discover new spam.

CHAPTER 5

METHODOLOGY

Using the initial analysis of collected tweets and observed behavior I have devised a methodology to capture spam accounts, given their changing behavior. I have developed new criteria to determine which accounts should be labeled as spam. The following section presents the methodology used and the reasoning for the techniques behind it.

5.1 Discovering Spammers

Tweets that contain spam, and accounts that tweet spam can be very versatile in their behavior. However there are certain patterns that can be recognized given an in depth analysis. Based on initial analysis and observations of tweets, I came to the conclusion that in order to discover spam we need to do the following:

1. Initial labeling of a tweet as spam or benign
2. Detection algorithm to determine the spam score.

The spam score represents the likelihood of a tweet or an account to be spam. The following subsection will present in detail each of the steps mentioned above.

5.1.1 Initial labeling as spam

The spam discovery algorithm explained in this subsection will run based on initial spam. The initial spam tweets discovered, will be used in the computation to detect further spam tweets and label accounts as harmful. In order to label a tweet as initial spam, multiple steps need to be performed. For each account (user) the major steps are:

- Iterate through each tweets

- Follow the shortened URL
- Decide whether URL has been flagged (crawl website)
- Label tweet as initial spam or not spam

Twitter has its own mechanism to decide whether a website is spam or not. I label a website as spam if Twitter or other used URL shortening web service(Google, bit.ly etc.) flagged it as such. A flagged website will prompt the user before displaying the end link. An example of a flagged URL can be found in Figure 1. I implemented a web crawler to check if the short URL leads to a warning page. The web crawler looks for the word "Warning" on the first web-page displayed. If the word is encountered, the tweet gets marked as spam and assigned an initial spam value of 1, otherwise assigned a value of 0. If a tweet is initially labeled as non-spam (with a spam score of zero) it does not mean that it isn't, but simply means it may not have been flagged yet, or indeed it isn't spam. The final decision of whether a tweet u is spam or not will be made after further analysis of the user account(that tweeted u) and the tweets of the other user accounts, as well as the interaction among them. It is important to check whether other accounts have tweeted the same tweet u, because if those accounts are spam, it is highly likely for this tweet to be spam as well. Spam accounts tend to tweet the same, or very similar tweets as shown in Table 3. The final decision(score of a tweet) is a likelihood of the tweet to be spam.

However, in order to discover similar spam tweets, assigning an initial spam value=1 to all tweets with a flagged URL is insufficient. I make use of another method that involves matching tweets by pattern. It starts by creating a hash value for each tweet and comparing it to the hash value of other tweets. The initial analysis of the tweet pool shows that simply hashing the entire tweet is insufficient as spammers tend to alter the tweet by using different characters or digits. For this reason, each individual tweet, will be stripped of all non-alpha numberic characters such as digits 0-9 or characters like *,!,@,#. Furthermore any link that starts with http or https

will be removed from the tweet, as well as any mentions of other users that start with @user or any hashtags that start with #hashtag.

The following table illustrated how spam accounts use different mentions to target a particular audience using the same exact tweet:

Table 4.: Spam usage of different mentions

Tweet
@Lorin_Marie Make An Incredible Income - Follow The Simple Steps http://t.co/NhghOoSJ
@lovely_lauren19 Make An Incredible Income - Follow The Simple Steps http://t.co/NpqkGerf
@DrTiaCMTyree How to Make Money on the Internet http://t.co/NhghOoSJ
@TheOaklandPress How to Make Money on the Internet http://t.co/NhghOoSJ
@stargaryen How to Make Money on the Internet http://t.co/Evq7uBT0

This insures that similar tweeting patterns will be recognized among the tweet pool. Specifically I am interested in matching hashes of tweets that have been initially labeled as spam. This method allows the algorithm to discover malicious tweets that might contain different links, different mentions or characters etc, but are at core similar to an already flagged tweet. This ensures the discovery of tweets that have not yet been labeled as spam. It turns out that if a tweet u_j has the same hash as a malicious tweet u_{j+1}, the likelihood of that tweet u_j to be spam will be significantly increased. For this reason I set all initial spam scores of tweets that have the same hash with an initial flagged tweet equal to 1.

The following table show examples of tweets making use of different numbers, links and hashtags but after the tweets is simplified, similar core tweets get the same hash value:

Table 5.: Hashed Tweets

Tweet	MD5 Hash
Best diet pill to lose 30 pounds in 1 month! http://t.co/X8hl8A1H	3b93dce5649dc0cf3a719a8384b12575
Best diet pill to lose 30 pounds in 1 month! http://t.co/rnqajwUX #fit	3b93dce5649dc0cf3a719a8384b12575
Best diet pill to lose 30 pounds in 1 month! http://t.co/Os1EV74c	3b93dce5649dc0cf3a719a8384b12575
The Best diet pill to lose 30 pounds in 1 month! http://t.co/u52sPIle	17ec8f846fd977289caaa3f626ad2986
The Best diet pill to lose 20 pounds in 1 month! http://t.co/DLJBMz9n	17ec8f846fd977289caaa3f626ad2986

5.1.2 Spam Likelihood Calculation

Using the link crawling and tweet hashing methods described in the above subsections, I devise a mathematical formula that uses the initially assigned spam values of tweets, to assign a final spam value to all tweets and users. It was obvious to me from an initial analysis of the tweets and accounts, that there is a correlation between spam tweets and accounts. Whether this correlation is repeated spam tweets or similar tweeting patterns among spam accounts, it is important to use this correlation to assign a final spam value to tweets and users.

The mathematical formula determines how likely it is for a user or a non-flagged URL to be malicious. It assigns a value between 0 and 1 to each user and tweet. These values are store in 2 vectors X(representing all the users spam scores) and U(representing all the tweets spam scores).

The following assumptions stand:

1. The more malicious URLs a user tweets, the more likely it is for the user to be a malicious account.

2. The more a URL is tweeted by malicious users, the more likely it is for the URL to be spam.

Used notations:

Table 6.: Notations

Notation	Explanation
$X := \{x_0, x_1, ...x_n\}$	the spamming scores for n users where:
	$x_i \epsilon$ [0,1] representing the likelihood of a user to be malicious.
	n is the total number of distinct users that have tweeted a URL containing tweet
$U := \{u_0, u_1, ...u_m\}$	the spamming scores for all the m URLs tweeted by users where:
	$u_j \epsilon$ [0,1] representing the likelihood of a URL to be malicious.
	m is the total number of distinct URLs tweeted
x_i^0, x_i^t	the initial value of x_i at round 0 and the value of x_i at round t respectively
u_j^0, u_j^t	the initial value of u_j at round 0 and the value of u_j at round t respectively
$N(i)$	set of tweets of user i
$N(j)$	set of users that have tweeted URL j
α and β	predetermined constants, $\alpha, \beta \epsilon$ [0,1] such that $\alpha + \beta <1$
ϵ	ϵ is a sufficiently small constant where the computation converges

Initially all elements of the set X are equal to 0, which means that all users are considered to be non-malicious. At the end of the algorithm, each element in this set will be equal to a real number $x_i \epsilon$ [0,1] which represents the likelihood of being a malicious user. On the other hand, all elements in the set U will initially be either 0 or 1. The elements in this set that have been set to equal 1, are the URLs that have already been flagged as malicious by Twitters defense system or by the hashing method described in the previous section. All other URLs in the set U associated with the value 0, have not yet been classified as being malicious or truthful. Having a value of 0, does not equate to being truthful, it means it has yet to be classified.

The first step presented in the following formula, assigns the initial spam value to all users in

X and all tweets in U: Step 0: $x_i^0 = 0 \; \forall i$ and

$$u_j^0 = \begin{cases} 1 & \text{if URL } j \text{ was flagged} \\ 0 & \text{otherwise} \end{cases}$$

At the next iteration(Step t) and until the algorithm converges each user and tweet will be assigned a different value at each step t, based on the interaction of the malicious users with non-malicious users and common tweets among them. Analogically at each step t+1 each tweet or user receives a different spam value based on the spam value at step t. Step t:

$$x_i^t = \alpha \frac{1}{|N(i)|} \sum_{i \epsilon N(i)} u_j^{t-1} + (1-\alpha) x_i^{t-1} \tag{5.1}$$

$$u_j^t = \alpha \frac{1}{|N(j)|} \sum_{j \epsilon N(j)} x_i^{t-1} + (1-\alpha-\beta) u_j^{t-1} + \beta u_j^0 \tag{5.2}$$

βu_j^0 used in order to assign a higher value to a URL that was initially marked as spam.

The computation converges where:

- $||U^t - U^{t-1}|| + ||X^t - X^{t-1}|| < \epsilon$

  $\epsilon = 0.001$

At each step t, for each user x_i^t the algorithm computes the spam sum at time t-1 of all the tweets user x_i has tweeted and adds the spam score for user x_i at the previous step t-1. Similarly for each tweet u_j^t, the algorithm computes the spam score off all the users x_i at time t-1 that have tweeted u_j, then adds the previous score of u_j at time t-1 while also adding βu_j^0 representing the initial spam score of this tweet. βu_j^0 ensures that tweets that have been initially flagged, receive a higher spam score. Because the algorithm iterates through each index of the X vector while looking at all indices of the U vector, the algorithm complexity will be $\Theta(N * M)$ where N is the total number of users and M is the total number of tweets. The next step in the algorithm is to iterate through each index in the U vector and for each index look at all indices in the X vector. This results in a complexity of $\Theta(M * N)$. The final complexity of the algorithm is

$\Theta(N * M) + \Theta(M * N)$

The described algorithm will be applied to the database of tweets and users and when it terminates there will be a spam likelihood score associated with each account and each tweet. Do determine whether a user or tweet is spam, I select a threshold and say that the user or tweet with a spam score above the threshold is spam. This threshold was chosen by analyzing the histograms of the tweets and users spam score distribution at each run of the algorithm with a different α and β score.

In this chapter, I presented 2 different methods incorporated in the Spam Discovering Algorithm presented in Chapter 3, the initial labeling of spam and the mathematical formula that uses the initial spam scores (0 or 1) to assign the final spam score. The next chapter will present the results of the algorithm run, as well as the experiment set up and the algorithm validation.

CHAPTER 6

RESULTS

Last section of the previous chapter introduces an algorithm to determine how likely it is for a tweet or a user to be spam. The results of the algorithm has assigned tweets and user to be either not likely or very likely spam with a final spam score as high as 0.9999. These results show that initially flagged have a very high chance to represent spam. In the following sections, I introduce more results and an in depth analysis of the scores obtained by the spam likelihood calculation as well as the experiment setup. I intend to design the experiment following the list below:

1. I am interested to know the spam scores for all accounts and tweets and initially labeled tweets, after convergence.

2. I would like to determine the appropriate parameter setting for the score computation algorithm.

3. I would also like to evaluate the effectiveness of the score computation algorithm.

In the following subsections I will describe the experimental set up followed by the experimental results.

6.1 Experiment Set Up

This section will give an overview of the pre-requisites used for the experiment as well as the experiment set up. In order to run the algorithm on the data set, the data needs to be partitioned so that it allows the experiment to run multiple times with different parameters, without corrupting the initial data. I made a copy of the database while backing up the initial

one to maintain the integrity of the collected data. The experiments were conducted by running bash shell scripts on the available server. The bash shell script uses different java files as well as the R language [18]. The java code was used to store complex data structures such as hash-sets and hash-maps and permit faster manipulation of the data set. Bash was used for the database interaction while the R language was used for statistical purposes, such as extracting the median, standard deviation, mean, max and min or different results. The algorithm requires different parameters, such as α, β and ϵ. The following table showd the different values α, β and ϵ as well as the number of iterations it took to converge, the number of users assigned a spam score higher than 0.1 and the X and U vector distance (from X and U when $\alpha = 0.1$ and $\beta = 0.2$):

Table 7.: Vector Distance and Iterations

α	β	Iterations	Nr. users>0.1	X distance	U distance
0.1	0.1	217	108	1.42E-5	1.82E-4
0.1	0.2	144	109	-	-
0.1	0.5	101	109	9.23E-6	1.81E-4
0.1	0.8	89	109	1.16E-5	2.42E-4
0.2	0.1	194	107	3.97E-5	3.66E-4
0.2	0.2	116	108	1.42E-5	1.82E-4
0.2	0.5	68	109	3.10E-6	5.20E-5
0.5	0.1	181	104	1.00E-4	5.56E-4
0.5	0.2	101	104	5.13E-5	4.19E-4
0.8	0.1	178	100	1.48E-4	6.22E-4

α and β affect the convergence of the algorithm. A smaller α and a smaller β results in a higher number of iterations. When α increases from 0.1 to 0.2 and β increases from 0.2 to 0.5

the number of iterations decreases significantly. Another observation is the fact that the spam values assigned to a tweet or an account is either much higher(if it is spam) or much lower(if not spam) when α and β are smaller. This is due to the fact that the number of interaction increase as α and β decrease.

The convergence of the U and X vectors seen in the following figures. The figures represent the algorithm results with $\alpha = 0.2$ and $\beta = 0.2$. The numbers were calculated by measuring the vector distance at each iteration. The y-axis represents the vector distance at a particular iteration while the the x-axis represents the number of iteration.

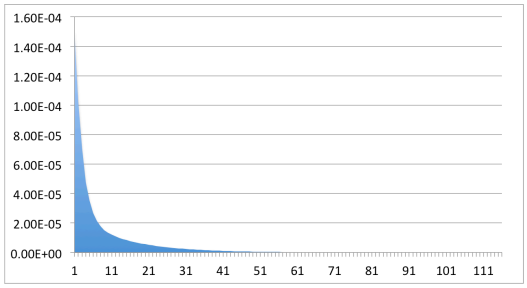

Fig. 5.: U vector distance-Convergence

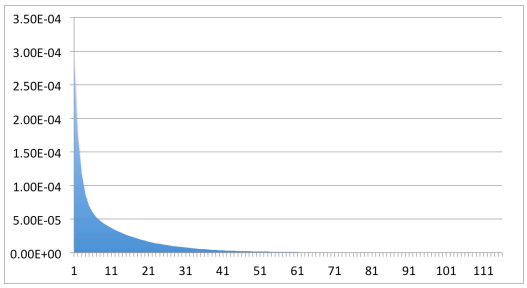

Fig. 6.: X vector distance-Convergence

An abrupt decline in the distance can be observed from the beginning up to the 11th iteration, after which the algorithm slowly reaches convergence.

6.2 Score Distribution and Algorithm Parameters Set-Up

The following graph illustrates the distribution of the spam score vector U during different runs.

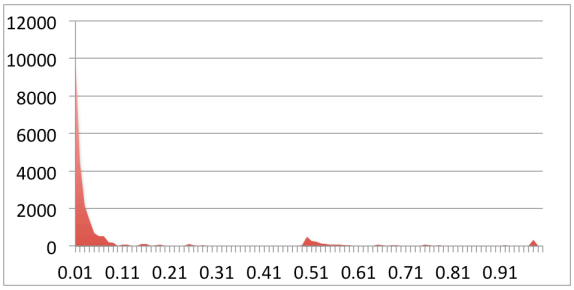

Fig. 7.: Tweets spam score distribution $\alpha = 0.1, \beta = 0.1$

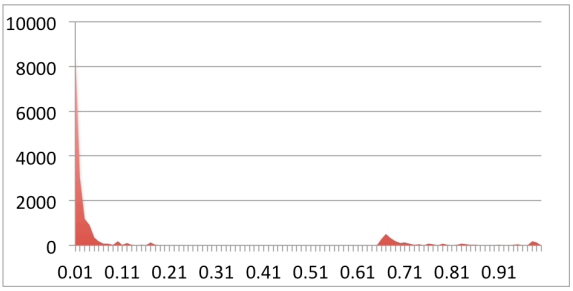

Fig. 8.: Tweet spam score distribution $\alpha = 0.1, \beta = 0.2$

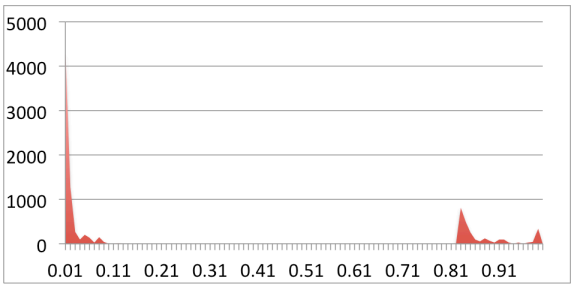

Fig. 9.: Tweet spam score distribution $\alpha = 0.1, \beta = 0.5$

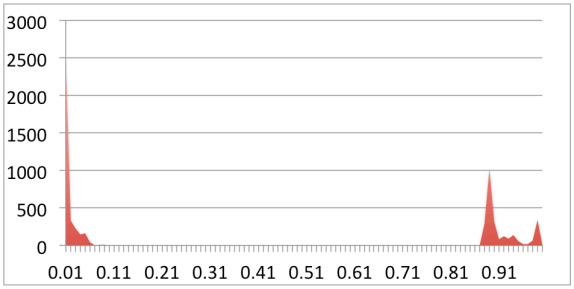

Fig. 10.: Tweet spam score distribution $\alpha = 0.1, \beta = 0.8$

The vector U contains 3.5M tweets and each index in the vector represents the spam score (for a tweet). When β is large most of the spam scores are within the [0.0-0.1] and [0.8-0.99] range. This is because in the mathematical equation used to discover spam, βu_j^0 will keep the initial spam score of the tweet. From these graphs, the following assertions can be made:

- A tweet with a score below 0.1 is not spam
- A tweet with a score above 0.8 is highly likely spam

It is interesting to observed that all the initial spam tweets that were labeled as malicious by Twitter (or other URL shortening tools) have received a final score of 0.65 or higher. This distribution of spam scores for initially flagged tweets can be seen below: All initially flagged tweets received a final spam score of 0.65 and higher.

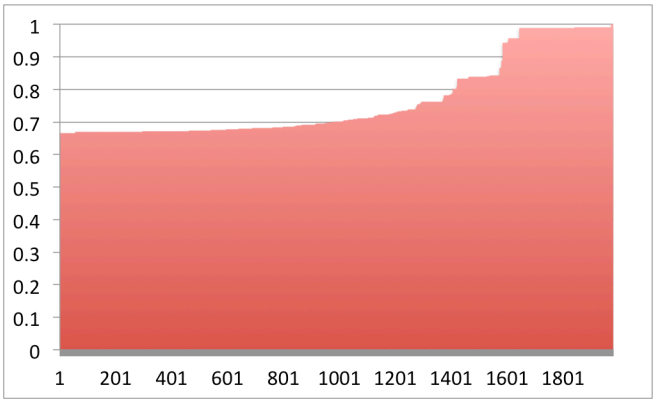

Fig. 11.: Initial Spam Tweet Final Spam Score Distribution

6.3 Experimental Results

Given an initial set of spamming tweets, users that have tweeted multiple flagged tweets, have resulted with a high spam likelihood score. The majority of these tweets are spam, and easy to spot. It is not surprising that the following tweets have received a spam score as high as 0.9999:

- @mildsto*** Real ways to make money using computers and the Internet http://t.co/NhghOoSJ
- want to start your own business in 2013? look at this - http://t.co/kJIAtUjm

These tweets are obvious spam and expected to have a high spam score. Similar tweets that were initially not flagged but have been tweeted by many spam accounts are also expected to have a high spam score. The following tweets are examples of obvious spam, which were not flagged by Twitter, but were discovered as spam:

- My best week! Earned $231.35 doing surveys in past week :) LOOK » http://t.co/f6dTPIFk
- Just downloaded the Webs Best Investment Sites magazine by Old School Value Jae_Jun http://t.co/zx158HTHMl

By analyzing tweets that received a high spam score, there are certain patterns that can be observed. Most of these tweets give an incentive to the user to click on the link. It is interest-

ing to see the similar pattern/phrasing they use. A popular trending topic observed is dietary advice, mostly the promise to lose weight by using a certain product or following a diet or promotion of certain weight loss product such as raspberry ketones or green coffee beans.

6.4 Algorithm Validation

In the following subsection, the accuracy of the spam likelihood algorithm will be calculated using the K-Fold cross validation [4]. Cross validation is a technique used primarily in statistics for estimating the performance of a predictive model. The validation model uses a data set for training, as well as a set of data against which the model is tested. The validation technique uses the original data sample to partition it into k randomly chosen subsets. Out of these k subsets, 1 is used for validation while the other k-1 sets are used as training data. The process repeats k times, so that each k set is used once as the validation.

For the spam likelihood algorithm, the data set used contains all initially flagged spam tweets. This data is stored in an indexed table, which is partitioned into k sets. K is chosen to be 10, dividing the data set in 10 sets. All tweets in this table contain an initial spam score=1.

Table 8.: Notations

Notation	Explanation
μ	Mean value of the spam likelihood values
$\widetilde{u}$	Median value of the spam likelihood values
max	Maximum value among spam likelihood values
min	Minimun values among spam likelihood value
σ	The standard deviation of the spam likelihood values

The Ten-Fold cross validation results(column 5 below) proves that the algorithm is efficient at discovering new spam tweets by learning from the existing spam tweets in the data set.

Table 9.: Ten-Fold Statistics

	μ	*max*	*min*	%>0.3
0-10%	0.7264	0.9899	0.625	88%
10%-20%	0.7398	0.9881	0.625	92.5%
20%-30%	0.7167	0.9999	0.625	88.5 %
30%-40%	0.7422	0.9999	0.625	91%
40%-50%	0.7112	0.9999	0.3245	91.5%
50%-60%	0.7112	0.9899	0.625	91%
60%-70%	0.9899	0.7440	0.0625	91.5%
70%-80%	0.7371	0.9998	0.1490	86%
80%-90%	0.7318	0.9899	0.625	90%
90%-100%	0.7455	0.9999	0.625	92.9%

6.5 Evaluation

The preliminary results show that all tweets flagged as spam have a spam likelihood value of 0.625 (or higher) whereas the maximum value is 0.9999976. The average spam likelihood of these tweets is μ=0.7356384 and the median is 0.6641736.

For the tweets that were initially not labeled as spam by Twitter or another URL shortening web-service the algorithm start with a initial spam value of 0. At the end of the iterations the algorithm has assigned the maximum spam value of 0.3649912 and the minimum value of 0. The mean of these values is μ=0.0001477763 while the median is 4.5e-09.

Some statistics of the algorithm results:

Table 10.: Statistics

	μ	$\widetilde{u}$	max	min	σ
All tweets	0.0005604816	4.5e-09	0.9999976	0	0.01787612
spam=1	0.7356384	0.6641736	0.9999976	0.625	0.1351345
spam=0	0.0001477763	4.5e-09	0.3649912	0	0.002438559
All users	0.0005593829	1.17e-08	0.9999949	0	0.01283942

To verify the accuracy of the spam detection algorithm I manually evaluated the highest ranked spam accounts (initially not flagged by Twitter). To verify whether a tweet is indeed spam or not, I checked the URL embedded in the tweet.

Out of 100 highest ranked tweets:

- 89 URLs were spam (either blocked or dead).
- 11 URLs were still alive.

Out of 200 highest ranked tweets:

- All tweets were spam.
- 172 URLs were either blocked or dead.
- 28 URLs were accessible.

Analyzing the URL's that were accessible(172), I was able to conclude that all of them can be categorized as spam. They represent different websites that appear to be aggressive advertisement, with links to a myriad of online social websites, peer-to-peer websites and different product endorsement. For the safety of the readers due to their highly sexual content, I decided not to post any of such examples.

Tweets with the highest scores that were not initially ranked as spam:

- A tweet composed of a link only to the highly popular(440M+ views) Youtube video of the Original Gymmy Bear song
- The 2nd highest is a tweet composed on only a link that has been blocked by Google
- Tweet: ”SenFeinstein as one of your constituents, I ask that you support H.R.6480 and S.3609 IRFA: http://t.co/ndOo3x8l #FairNetRadio ”

It came as no surprise, that the highly popular video link, of the Youtube video of the Original Gummy Bear song, was a false positive. It is well recognized that spamming accounts will make us of popular topics, hashtags and links to try and reach a broader audience. For this reason, it is very hard to exclude and discover similar false positives.

Tweets that were initially marked as spam:

- check this out! I made almost $600 today so far http://t.co/m4e1PvU
- hey everyone youve got to check this out I made almost $500 today! http://t.co/ZURqqrk
- Walk out of your crappy 9-5 job this week! http://t.co/1XjQUr9t
- http://t.co/CeWjdnTcVW I could make some serious money selling nude pics of myself to bulimics with short fingers.
- I’m gonna start my own TV network called RealityTV(RTV) and play nothing but music videos http://t.co/R2CBvXXyIZ

An analysis of the tweets database shows that 4262 tweets have been labeled as spam with a spam value greater than 0.1. Out of these 4262, 1991 were initially labeled as spam and the rest had an initial spam score=0; An example of such tweets:

- Fastest way to lose weight, burn fat and stop fat production http://t.co/QXx6JQ2WCX
- Get Back Some Financial Independence Into Your Life. http://t.co/hZxh9GSd

- Make An Incredible Income - Follow The Simple Steps http://t.co/4dgsLKUL

These tweets display a similar pattern with the observed initial flagged tweets. They try to persuade the reader to follow a fast way to lose weight, making more money etc. I can conclude that the algorithm discovered an additional 2271 tweets, that were initially not labeled as spam. These tweets are associated with 1830 distinct users.

Below is a graph, showcasing the distribution of the tweets with a spam score 0.1(inclusive) or higher. It can be observed that about 1500 of the total amount of these tweets have a final

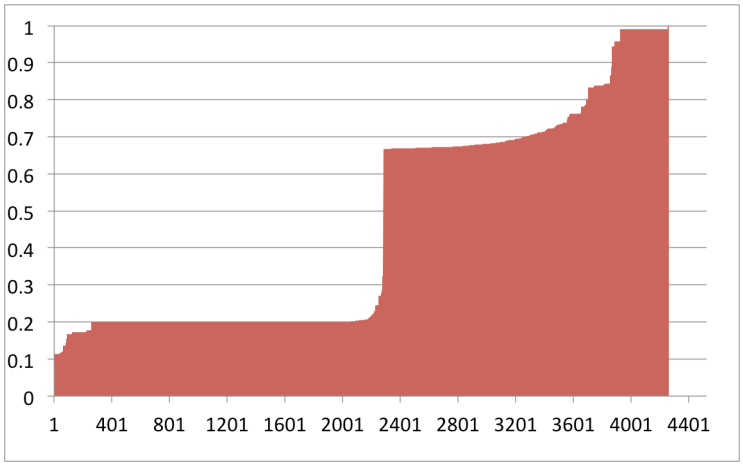

Fig. 12.: Tweets Spam Distribution

spam score of about 0.2. Also, there is a slow increase trend between 0.65 and 0.85, then about 400 of them with a score very close to 1. This similar trend of spike, can be observed in Fig.4 through Fig. 7. This is an indication that the algorithm tends to assign either a low score or high score to tweets. Not many tweets can be observed in the range [0.25-0.6] regardless of α and β. Most users that tweet spam tweets, either tweet very few or many spam tweets. The user that tweet very few spam tweets, tend to be compromised accounts for a short period of time. However most of the spam tweets that they tweet, are usually also tweeted by spam accounts. If a tweet is tweeted by a spam account and a honest account at the same time, the tweet will be assigned a high spam score. If a tweet is tweeted by only an honest account (that tweets a very

small spam to non-spam ratio), but not any spam account, the tweet will finally be assigned a spam score below the 0.1 threshold. However, if a tweet is tweeted by many spam accounts,it will be assigned a high spam score, above 0.6 regardless whether an honest user also tweeted it or not.

Analyzing the users that have been ranked with a score higher than 0.1, 108 of them were finally marked as spam. Manual analysis of the 108 highest ranked accounts shows that 102 are spam accounts while 6 seem to non-spam, legit users. Looking at the tweeting behavior of the 102 spam accounts:

- About 30% of them do not have a picture.
- Most of them have 1 picture in the profile.
- While some did have more than 1 picture in the profile, the pictures were non-personal (not a person) and also not related to each other.
- They often re-tweet every 8-10 tweets.
- Most had at least 1 exact tweet re-tweeted, where no words were changed. Most of the time this tweet was immediately repeated.
- They all included a pattern of tweeting by starting the tweet with a mention.
- Inconclusive results can be drawn from the number of followers or friends they have, since the number were ranging from a handful to a couple thousands.

CHAPTER 7

CONCLUSION AND FUTURE WORK

I presented a novel approach to discover spam accounts on Twitter, by analyzing the connection between accounts and the tweeting patterns of initial spammers. This approach aids in the discovery of spam accounts and tweets. The algorithm presented proves to be efficient at discovering new spam tweets, using the tweeting pattern and behavior of spam accounts. The algorithm for spam discovery presented can sustain changes in tweeting patterns and account behavior, because it relies on the interaction among spam accounts and the usage of flagged tweets.

For future work I would cluster the discovered accounts based on the spam discovery algorithm presented in Chapter 5. A multi-level clustering algorithm can be efficient at clustering these accounts. An initial run of this algorithm has been made, but was unable to draw conclusions, since the spam likelihood calculation method was not developed. Some patterns of similar spam have been observed, such as the incentive to make money from home, lose weight and buy different Apple product (iPads specifically), at this time, I cannot determine that these patterns represent the same cluster.

CHAPTER 8

APPENDIX A

```
import java.io.\*;
import java.lang.\*;
import twitter4j.\*;//API package
public class GETTweet{
        private final static String CONSUMER_KEY = "7
           EDBioM2CN4GO753HT****";
    private final static String CONSUMER_KEY_SECRET = "
       IV4MhXXaoPFF06ce8PvWhjcWL58bJmZawAmafL****";
    Database db = new Database();
    RateLimitStatus status;
    Map<String ,RateLimitStatus> rateLimitStatus;
    RateLimitStatus event;
    String endpoint;
    long currentID;

        public void start() throws TwitterException,
           IOException
                try{
        db.createConnection();
        }catch(Exception e) {System.out.println(e);}
```

```
ConfigurationBuilder cb = new ConfigurationBuilder()
   ;
    cb.setDebugEnabled(true)
  .setUseSSL(true)
      .setOAuthConsumerKey(CONSUMER_KEY)
      .setOAuthConsumerSecret(CONSUMER_KEY_SECRET)
      .setOAuthAccessToken("2272744616-0
         IKZO0DiQddyZC7tMphRivneiCVods7ikLP****")
  .setUser("******")
  .setPassword("******")
      .setOAuthAccessTokenSecret("9
         qFp3lpu89Depb7zzfRmUl1Dq9nqAaFr5RqdSVeYz
         ****");

    TwitterFactory tf = new TwitterFactory(cb.build
       ());

 Twitter twitter = new TwitterFactory().getInstance
    ();
 twitter.setOAuthConsumer(CONSUMER_KEY,
    CONSUMER_KEY_SECRET);
 String accessToken = getSavedAccessToken();
 String accessTokenSecret =
    getSavedAccessTokenSecret();
```

```
AccessToken oathAccessToken = new AccessToken(
   accessToken,
 accessTokenSecret);
twitter.setOAuthAccessToken(oathAccessToken);
twitter.addRateLimitStatusListener( new
   RateLimitStatusListener() {
@Override
public void onRateLimitStatus( RateLimitStatusEvent
    event ) {

   //puts the thread to sleep for 15min (900 sec)
      before limit is reached
   try{
   if(event.getRateLimitStatus().getRemaining()<2){
                System.out.println("␣\n␣sleeping␣\n
                   ␣");
               TimeUnit.SECONDS.sleep(900);
           }
   } catch(InterruptedException ex) {
       Thread.currentThread().interrupt();
   }

   System.out.println("Limit["+event.
      getRateLimitStatus().getLimit() + "],␣
      Remaining[" +event.getRateLimitStatus().
```

```
        getRemaining()+"]");
}
//end of limitation

@Override
public void onRateLimitReached( RateLimitStatusEvent
    event ) {
    System.out.println("Limit["+event.
        getRateLimitStatus().getLimit() + "], 
        Remaining[" +event.getRateLimitStatus().
        getRemaining()+"]");
}
} );
    ArrayList<Long> dbId= new ArrayList<Long>() ;
    long returned;
    try{
        dbId = db.getAllIDFromDB();
    }catch(Exception e){System.out.println(e);}

String screenName,text,date;
int follCount,friendsCount;
boolean isRetweet=false,isRetweeted=false;
User user;
try{
for(int i=0;i<51000;i++){/
```

```
currentID = dbId.get(i);
Paging paging = new Paging(1, 200);
String name = db.getName(currentID);
user=twitter.showUser(name);
boolean isProtected = twitter.showUser(name).
   isProtected();
if(isProtected)//if its not protected
    System.out.println( user.isProtected()+"\n"
       + "protected␣user" + "\n");
else{
    System.out.println("get␣status␣return␣is:␣"+
       user.getStatus());
ResponseList<Status> list2 = twitter.
   getUserTimeline(name, paging);
for (Status tweets : list2) {
    date = ""+tweets.getCreatedAt();
    text = tweets.getText();
    screenName = tweets.getUser().getScreenName
       ();
    follCount = tweets.getUser().
       getFollowersCount();
    friendsCount = tweets.getUser().
       getFriendsCount();
    isRetweet = tweets.isRetweet();
    isRetweeted = tweets.isRetweeted();
```

```
                if(isRetweet)
                db.addTweet(currentID,text,2,date);
            else if(isRetweeted)
                db.addTweet(currentID,text,3,date);
            else db.addTweet(currentID,text,1,date);
            }
        }
    }
}catch(Exception e){System.out.println(e);};
            private String getSavedAccessTokenSecret() {
         return "9qFp3lpu89Depb7zzfRmUl1Dq9nqAaFr5RqdSVeYz
            ****";
            }
            private String getSavedAccessToken() {
         return "2272744616-0
            IKZO0DiQddyZC7tMphRivneiCVods7ikLP****";
            }
            public static void main(String[] args) throws
               Exception {
         new GETTweet().start();
            }
}
   /
```

REFERENCES

[1] L. Alvisi, A. Clement, A. Epasto, S. Lattanzi, and A. Panconesi. Sok: The evolution of sybil defense via social networks. *2012 IEEE Symposium on Security and Privacy*, 0:382–396, 2013.

[2] M. Arrington. Odeo releases twttr, 2006.

[3] R. Beneito-Montagut, S. Anson, D. Shaw, and C. Brewster. Governmental social media use for emergency communication. In *Proceedings of the 10th International Conference on Information Systems for Crisis Response and Management, Baden-Baden, Germany*, 2013.

[4] Y. Bengio and Y. Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *The Journal of Machine Learning Research*, 5:1089–1105, 2004.

[5] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, NSDI'12, pages 15–15, Berkeley, CA, USA, 2012. USENIX Association.

[6] A. Cocotas. The state of social media avertising, and where it's headed. 2013.

[7] A. C. E. S. conducted by comScore. 2013 ups pulse of the online shoppertm. 2013.

[8] G. Danezis and P. Mittal. Sybilinfer: Detecting sybil nodes using social networks, 2009.

[9] B. J. Duggan M. The demographics of social media users — 2012. 2013.

[10] N. C. B. Insider. The real history of twitter, 2011. http://www.businessinsider.com/how-twitter-was-founded-2011-4.

[11] J. E. T. W. S. Journal. Inside a twitter robot factory, 2013. http://online.wsj.com/news/articles/SB10001424052702304607104579212122084821400.

[12] D. J. Kim and T. A. Yang. Social networking as a new trend in e-marketing. In *IFIP International Federation for Information Processing*, volume 255, 2010.

[13] J. D. B. Labs. Twitter underground economy still going strong, 2013. http://www.net-security.org/article.php?id=1859&p=1.

[14] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 1–9, New York, NY, USA, 2010. ACM.

[15] K. Thomas, C. Grier, D. Song, and V. Paxson. Suspended accounts in retrospect: An analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, pages 243–258, New York, NY, USA, 2011. ACM.

[16] K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson. Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse. In *Proceedings of the 22Nd USENIX Conference on Security*, SEC'13, pages 195–210, Berkeley, CA, USA, 2013. USENIX Association.

[17] I. A. f. w. t. S. Twitter and . Exchange Commission on October 15. Twitter inc. s1 form. 2013.

[18] W. N. Venables, D. M. Smith, R. D. C. Team, et al. An introduction to r, 2002.

[19] A. H. Wang. Don't follow me: Spam detection in twitter. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pages 1–10, July 2010.

[20] G. Wang, M. Mohanlal, C. Wilson, X. Wang, M. J. Metzger, H. Zheng, and B. Y. Zhao. Social turing tests: Crowdsourcing sybil detection. *CoRR*, abs/1205.3856, 2012.

[21] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai. Uncovering social network sybils in the wild. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, pages 259–268, New York, NY, USA, 2011. ACM.

[22] H. Yu, P. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 3–17, May 2008.

[23] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: Defending against sybil attacks via social networks. *SIGCOMM Comput. Commun. Rev.*, 36(4):267–278, Aug. 2006.