Noname manuscript No.
(will be inserted by the editor)

Discover Millions of Spammers in Weibo

Yi Zhang · Jianguo Lu

Abstract Weibo is the Chinese counterpart of Twitter that has attracted hundreds of millions of users. Just like other online social networks (hereafter OSNs), it has a large number of spammers that are created to boost the ranking or follower number of other accounts. Spammers are difficult to identify individually, especially when they are created by sophisticated programs or controlled by human beings directly. This paper proposes a novel spammer detection method that is based on the very purpose of the existence of the spammers: they are created to follow their targets en masse with high regularity, resulting in near-duplicate accounts that have similar sets of followers.

The discovery of near-duplicates is a challenging task for such a large network. Instead of calculating the Jaccard similarity among all the pairs on the original graph, we estimate the similarities among large accounts by taking a sample graph that contains one million random nodes. We find 395 near-duplicates. From such near-duplicates, we identify 12 millions of spammers (account for 4.56% of the total users) and 741 millions of spam links (account for 9.50% of the total edges). Furthermore, we characterize several typical structures of the spammers, cluster these spammers into 34 spammer producers, and analyze the main targets of these spammers.

1 Introduction

Fake OSN followers have become a multimillion dollar business. In Twitter, bogus followers are sold in large quantities ranging from thousands to millions [16]. Rampant spamming is encroaching on the normal social network, disrupting the platform for social communication and viral marketing. Users, as well as service providers, want to detect and remove the spammers [19] [6] [2].

It is difficult to distinguish between a spammer and a normal account individually [3]. A spammer account can look like a normal account, with normal screen name and profile picture. It can also send messages that are consistent with its account profile.

School of Computer Science, University of Windsor, 401 Sunset Avenue, Windsor, Canada.
E-mail: {zhang18f, jlu}@uwindsor.ca

Spammers can be sophisticated robot accounts, or even directly controlled by real human beings.

One trace that spammers can not hide is the very purpose of their existence, that is, they are created to be sold in large quantities to many customers. Consider the following scenario for a given spammer producer who owns N number of spammers. Let us assume that the spammer consumers have few real followers, or their real followers are negligible compared with the spammers they buy. If the producer sells all its N spammers to two customers, those two customers will have the same set of followers, i.e., they are near-duplicates. To reduce the risk of being detected, the producer may sell part of its spammers, say $N/2$ of them, instead of the all-out strategy. Still, when there are a large number of customers, two subsets of the sold spammers may overlap and generate near-duplicates.

Regardless the underlying mechanism of the generation of near-duplicates, its occurrence is highly unusual that defies statistic possibilities. We borrow the idea from plagiarism detection and near-duplicate web page detection [8]. Just as it is impossible for two long articles to share most of their phrases, two bloggers are unlikely to have almost the same large group of followers unless they are fake.

Discovering near-duplicates among hundreds of millions of accounts needs an efficient method. Numerous algorithms are proposed to find the near-duplicates among web pages, such as the MinHash algorithm that extracts a short representative 'fingerprint' for each web page [8]. Our new challenge is that the Weibo data in its entirety are not available. Instead, we can only use the Weibo API to call the service remotely over the Web.

We take the sampling approach to estimate the Jaccard similarities among top bloggers in Weibo. When the threshold value is 0.9, we find 395 near-duplicate bloggers, who are then clustered into 34 spammer groups using the Jaccard similarity. Each cluster corresponds to the producer of the spammers.

Next, we verify that the accounts in each cluster are indeed spammers by showing that they have some uncanny regularities. Some clusters of spammers are obvious. For instance, all the spammers have zero followers or zero messages, or the same number of out-links. In some clusters, all the spammers are cell phone users, or registered in the same city and the same month, indicating that they are created by simple programs. In other clusters, most attributes are normal, indicating that they are created by a sophisticated process. However, these tens of thousands of accounts have the same maximally-allowed out-degree and form a closely knit link farm. We have also manually checked and verified many suspected spammers, and found that our method is accurate.

One may argue that there could be near-duplicates by chance. In theory, we show in Section 2.3 that such chance is extremely small. In practice, we manually checked these clusters of spammers, and demonstrate in Section 3.1 that they are obvious spammers. Even if there were occasional false positive cases in future applications, this method remains to be a robust one as long as the spammer industry exists: spammer producers need to sell the links en masse in order to be profitable. When there are large customers, the occurrence of near-duplicates is inevitable.

Our method is not only accurate, but also effective in capturing the spammers. In total, 12 millions of spammers are uncovered, which account for 4.56% of the

total number of accounts in Weibo. These spammers generate 54 millions of spam links, which account for 9.50 % of the total edges in the user network. In contrast to our method, other approaches can only identify very small number of spammers. For instance, the approach described in [5] only found 41 thousand spammers, and it depends on the suspended-account list given by the service provider.

From these 34 clusters, four common structures of spammers are identified, ranging from the most simple complete bipartite graph, to very complex link farm that reciprocate links to each other. In addition to spammer sources, we also studied their targets, and identified the top 'polluted' normal accounts who receive most of the spam links.

The **main contributions** of the paper are: 1) discovered 12 millions of spammers, along with their structure, origins and targets; 2) proposed a novel method to identify the spammers. It can detect sophisticated spammers as long as they follow their targets en masse; 3) presented an innovative star sampling method that is tailored for the Weibo web interface and Jaccard similarity.

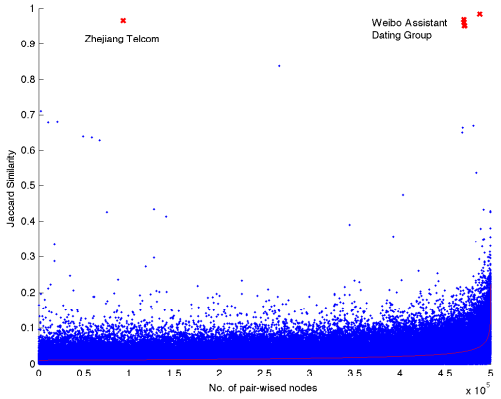
There are several implications of this paper. First, it heralds the end of the large-scale spam link industry, which sells links to large number of customers in large quantity. Second, more importantly, we demonstrate that those spammers could be identified using a small sample (0.5 % of the original user network) obtained from the web API, without the access to the entire data. OSN service providers have the motivation to keep their eyes closed on such market, to boost the total number of registered users. Using our method, a third independent party can spot such spammer groups. Thirdly, this is an success application that demonstrates the power of sampling methods. Normally, sampling methods are used to discover simple properties such as size [10] and average degree [4]. This paper shows that, by designing the sampling method carefully, interesting discoveries can be made using limited sampling interface. Lastly, we demonstrate that user network alone can be used to discover many spammers. Most of existing methods use many features, especially message patterns, to identify spammers. However, spammers can disguise their behaviour as normal accounts.

The remaining paper is organized as the following. Section 2 defines the near-duplicates, introduces our method to estimate the Jaccard similarity, and proves the accuracy of the estimation. Section 3 describes the properties of the spammers that are discovered, how they cluster the near-duplicates, therefore the spammers, into 34 groups. Section 4 discusses the spammer targets.

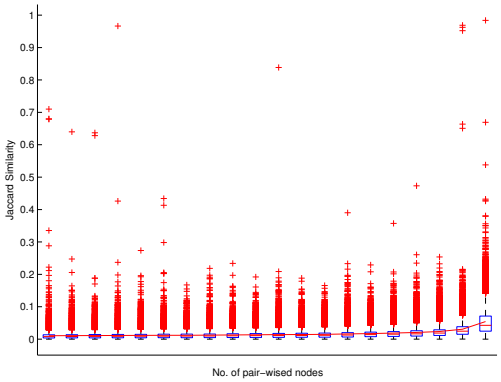
2 Near duplicate accounts

2.1 Near duplicates

Two large accounts are unlikely to have the same set of followers. When two books have the same set of n-grams, we say that there is a plagiarism; When two web pages receive the same set of hyper-links en masse, there is a Web link farm; When two weibo accounts share thousands or even millions of followers, with uncanny regularity, they are most probably artificially engineered. Borrowing the concept of near-duplicate



(A) Scatter plot



(B) Corresponding boxplot

Fig. 1. Jaccard similarities between the top 700 accounts in Weibo, sorted by the harmonic mean of the degrees.

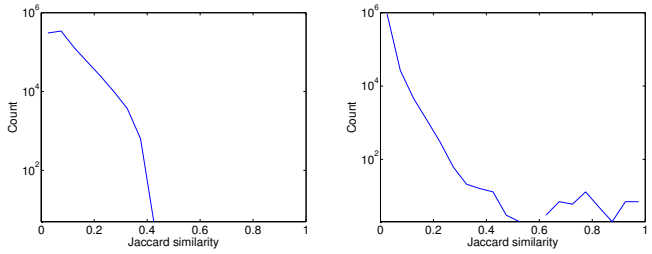
documents in information retrieval, we define the near-duplicate of OSN accounts as follows:

Definition 1 (Near-duplicates) *Two accounts a and b are called near-duplicates, denoted as $a \approx b$, if their Jaccard similarity in terms of their followers is close to one, i.e.,*

$$J_{ab} = \frac{|F(a) \cap F(b)|}{|F(a) \cup F(b)|} > \theta, \tag{1}$$

where $F(x)$ is the set of followers of account x, θ is a threshold value that is close to one. In our experiment we let $\theta = 0.9$.

For near-duplicates, we only consider large accounts, the accounts that have a large number of followers. In our experiment, we select the top 10,000 accounts, who have at least 50,000 followers. We obtain the Jaccard similarities among all the combination of these accounts, resulting in total about 5×10^7 pairs. Among them we



(A) Random graph (B) Weibo

Fig. 2. Jaccard similarity distributions of random graph (A) and Weibo (B). Bin-size = 0.05.

find 395 near-duplicate pairs. To demonstrate that these near-duplicates are outliers, Fig. 1 plots a part of these Jaccard similarities among the top 700 accounts, along with the expected values. Most of the observed Jaccard similarity are around the expected similarities, while there are a few outliers that are above 0.9. Those groups are WeiboAssitants, DatingGroup, and Zhejiang Telecom from right to left.

Fig. 2 Panel (B) gives a larger picture of Jaccard Similarity distribution among all Weibo accounts whose in-degree are larger than 600,000. As a comparison, the Jaccard similarity distribution of a random gram is plotted in Panel A. What is the same between these two graphs is that most pairs have very low similarities. In random graphs, the frequency decreases monotonically as a function of the Jaccard similarity. It converges to zero quickly, and the probability of having a Jaccard Similarity as high as 0.9 is close to zero. In Weibo, the frequency also drops to zero monotonically. What is surprising is the peak for high similarity values, which is most probably caused by spammers.

2.2 Discover Near-duplicate By Star Sampling

It is impossible to compute the Jaccard similarity directly. Weibo had over 200 millions of accounts when we sampled the data in 2011. Calculating the pair-wise combination between all the accounts is out of the question. Many accounts have very large number of followers, in the order of 10^7. Set intersection operation is costly for such large data. Although numerous efficient algorithms, such as MinHash [8], are proposed, they are all based on the assumption that the data in its entirety are available. In OSN application, data can be obtained only through web queries, which are costly because of the network traffic involved. Thereby, we use samples to estimate the Jaccard similarity.

A star subgraph is a node with all of its out-bound links. Star sampling is to take a subgraph that is formed by random stars. First, uniform random nodes are selected. Then, all the out-links of these nodes are collected.

Uniform random nodes are selected as follows: A uniformly distributed random number is generated within the range of $1, ..., 10^{10}$, and is tested whether it is a valid ID by probing Weibo web site. Overall, we found $n = 1.08$ million valid IDs, and they are uniform random samples regardless of the ID distribution. One may doubt the

randomness of the sample, by arguing that the IDs may not be randomly distributed across the ID space. We want emphasize that these IDs are random, and refer to Appendix in [7] for its proof. One way to understand it is that every range has the equal probability of being sampled. If a segment has less valid IDs, it will have less sample IDs. For instance, there are very few valid IDs below 10^8. Corresponding, there are proportionally less samples in this range.

In addition to the correctness of sampling, the bigger concern is its efficiency. The success of this sampling method is due to that: 1) Every Weibo account has an numeric ID, even when it has a screen name; 2) The ID space (10^{10}) is not large compared with the number of valid IDs (2×10^8) space; 3) Probing the validity of an ID is fast, and service providers do not impose limit on the number of times to such probe.

Once uniform random nodes are obtained, all the outbound links are extracted. For every remote call, service providers would return a small number of links. In addition, there are daily quota as for the number of calls allowed. Thus, this process would not be feasible if there were larger nodes that have many out-bound links, such as the case in Twitter. In Twitter, many accounts, such as Obama, contains millions of out-links. Thanks to the policy set by Weibo, all accounts can not have out-links exceeding 2000, with a few excepts that is slightly larger than that. Thus, all the out-links can be collected in our experiment.

These stars form a subgraph, where large accounts are sampled more often by the stars [20]. This subgraph can estimate the Jaccard Similarity of the original graph as explained below.

Given two accounts (nodes) n_1 and n_2. Suppose that their number of followers are D_1 and D_2, and their common neighbours are C. Suppose that the sample ratio is $p = n/N$, where n is the number of uniform random nodes in the sample graph, and N is the number of nodes in the original graph. In the subgraph, the expected number of common neighbours is $c = pC$, the expected number of degrees are $d_1 = pD_1$ and $d_2 = pD_2$, respectively. The Jaccard Similarity in the original graph is:

$$S = \frac{C}{D_1 + D_2 - C} \tag{2}$$

The similarity in the sample graph is

$$s = \frac{c}{d_1 + d_2 - c} = \frac{pC}{pD_1 + pD_2 - pC} = S \tag{3}$$

Thus, s is the unbiased estimator of S. Next we need to study how large is the variance. c, the common neighbours in the sample graph, follows binomial distribution B(C, p), whose expectation is

$$E(c) = pC. \tag{4}$$

According to the property of the binomial distribution, the variance of c is

$$var(c) = Cp(1 - p) \approx Cp. \tag{5}$$

The approximation holds when we assume that the sampling ratio is small. In our experiment, $p \approx 0.005$. The relative standard error (RSE) is

$$RSE(c) = \frac{1}{c}\sqrt{var(c)} \approx \frac{1}{\sqrt{c}}. \tag{6}$$

In our experiment, the minimal c is around 250, $RSE = 1/\sqrt{250} = 0.063$. Assuming that the distribution of c approximates a normal distribution, we can conclude that the 95% confidence interval for our estimation is within the range of $s \pm 0.126s$. Therefore the estimation has a high accuracy.

2.3 Suspected spammers

Near duplicates are suspicious in such circumstances, not only because they are obvious outliers as depicted in Fig. 1, but also because of the large population of Weibo accounts and large number of followers of the near duplicates. We draw an analogy to plagiarism detection. If two documents are short, they could be the same by chance. But long documents can be hardly the same by chance. Therefore, in our experiment we only consider large near-duplicates, the account that at least contain 50,000 followers. In plagiarism detection, Jaccard similarity is between shingles instead of terms, because the vocabulary is not large enough. In our setting, the total number of account are very large. There are around 2×10^8 accounts in Weibo. Given two accounts a and b that each account has 10^5 followers. When the followers are created randomly, the expected number of duplicates among the followers of a and b is

$$duplicates = \frac{sizeOfSubset1 \times sizeOfSubset2}{totalPopulation} \tag{7}$$

$$= \frac{10^5 \times 10^5}{2 \times 10^8} \tag{8}$$

$$= 50, \tag{9}$$

according to the classic capture-recapture model [11]. Hence, the expected Jaccard similarity is

$$50/(10^5 + 10^5 - 50) \approx 0.00025. \tag{10}$$

When we see a Jaccard similarity that is close to one, higher than the expected value by a factor of thousands, we have a reason to hypothesize that the links are manipulated deliberately. We call those accounts that follow at least two near-duplicates are suspected spammers (hereafter spammers for conciseness).

To gain confidence that those unusual accounts are indeed spammers, let us look into the details of one of the spammer group, the Antique shops, as illustrated in Fig. 3. More details of these Weibo accounts, as well as other groups, can be found at `http://cs.uwindsor.ca/~jlu/spammer`. This group has 24 near-duplicates, all of them have the same followers in the amount of 0.2 million. Among these 0.2 million spammers, 96 percent have no followers at all; 81.68 percent have the

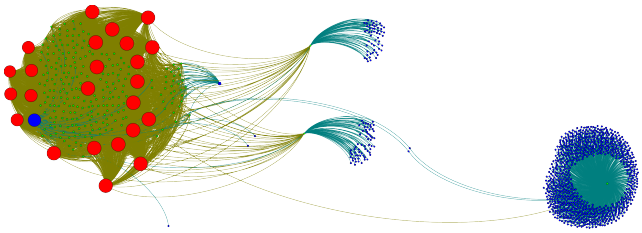


Fig. 3. The antique shop group: near-duplicates (the red nodes) are followed by the same group of spammers (green nodes). Those spammers are mostly of zero incoming links. Most of the spammers point to the near-duplicates only, with a few exceptions that point to a larger number of other nodes (the blue nodes).

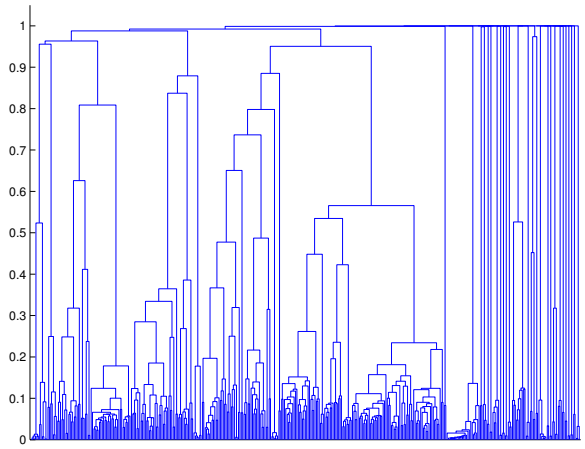


Fig. 4. Dendrogram of the clustering result of the 395 near-duplicates.

same out-degree (25, pointing to the near-duplicates only); 86 percent are registered on the same two months (April and May, 2011); 80.06 percent are registered from Beijing; zero percent are cell phone users or verified users; 95.57 percent have never sent any messages. Such high regularity clearly demonstrates that they are created deliberately to boost the follower number.

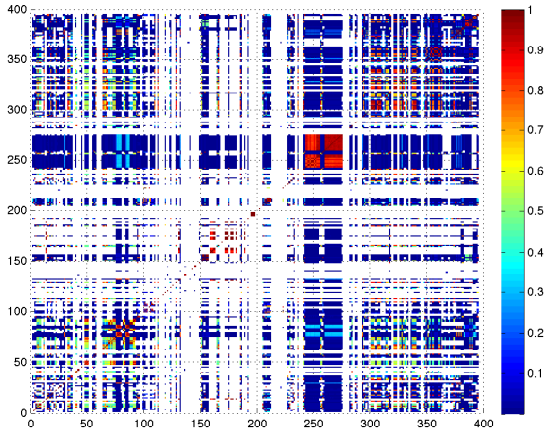
While the Antique shop is a typical spamming structure, we find a variety of spamming topologies, ranging from a complete bipartite graph to sophisticated follower link farms, the same as the link farm in the web [21]. We will discuss these structures in Section 3.1.In addition, We have manually checked these 395 near-duplicates, and found that 138 of them are already suspended, indicating their participation in spamming activities. For suspected spammer accounts, the number is too large to check them exhaustively. Among 100 randomly selected suspected accounts, we find that 95 are spammers (13 suspended, 14 highjacked), and 5 are legitimate.

Cluster	Spammers				#Near-duplicates	Name
	In-deg (avg)	Out-deg (avg)	#Spammers ($\times 10^6$)	#Links $\times 10^6$		
1	0.00	2.00	0.31	0.64	2	Love Shopping
2	0.00	4.00	0.06	0.27	2	Android Group
3	0.00	4.01	0.06	0.26	2	Gif Animation
4	0.17	5.73	0.07	0.44	3	Campus Chongqing
5	0.20	5.62	0.10	0.61	4	Campus Shangrao
6	0.43	5.69	0.22	1.30	3	Spam Group
7	0.84	7.29	0.09	0.70	2	Campus Jinan
8	0.95	6.09	0.11	0.71	2	Campus Xian
9	1.01	4.11	0.24	1.01	2	Mobile Neimengu
10	1.33	6.58	0.62	4.09	2	Mobile Winner
11	2.85	8.88	0.27	2.42	6	Liaoning Telecom
12	3.40	56.24	0.10	5.83	4	3G
13	3.58	6.79	0.86	5.84	2	Zhejiang Telecom
14	4.38	22.19	0.09	2.07	2	Mobile Dream
15	5.29	4.66	0.22	1.02	2	Love Hubei
16	7.15	18.96	3.49	66.24	2	Weibo Assistant
17	9.12	7.55	0.13	0.95	2	Mobile Marketing
18	10.22	69.19	2.49	172.71	3	Dating Group
19	10.68	36.43	0.65	23.63	5	Telecom Animation
20	15.51	29.41	0.20	5.96	24	Antique Shop
21	16.91	7.69	0.22	1.68	2	Telecom Jilin
22	23.97	17.53	0.05	0.95	3	Telecom Wuhan
23	41.56	213.27	0.08	17.49	7	Naming
24	44.55	99.14	0.12	12.05	2	Photo
25	47.58	130.10	0.16	20.19	12	Deleted
26	60.30	675.34	0.28	190.22	55	Green Tea etc.
27	63.54	204.11	0.27	42.18	53	Pets etc.
28	67.84	270.69	0.19	52.11	45	Wedding etc.
29	78.02	333.40	0.07	23.24	6	Deleted
30	93.05	924.70	0.19	177.81	117	Sydney Coupon
31	125.48	484.12	0.06	28.38	2	Spam Farm
32	133.09	134.46	0.07	8.68	3	Software
33	203.13	615.68	0.07	42.03	2	Health
34	251.16	504.65	0.08	43.55	10	Chinese med
Sum			11.90	741.10	395	
Mean	14.01	61.83				

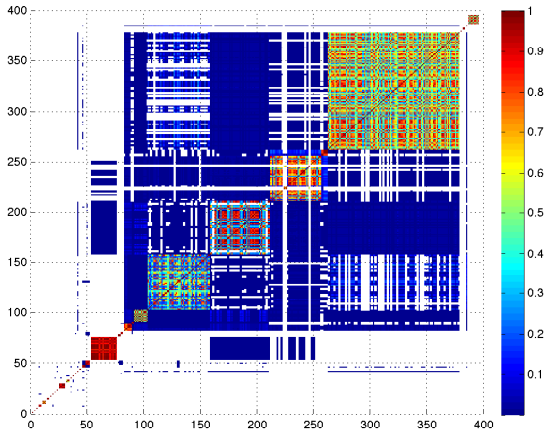
Table 1. 34 Clusters of near-duplicates and the statistics of their spammers, sorted in the increasing order of their average in-degrees. Each cluster has a clickable URL link that points to our web page showing the details of the near-duplicates and their similar accounts.

3 Clusters of Near-duplicates

When the threshold value $\theta = 0.9$, we find that there are 395 near-duplicate accounts in Weibo in 2011. From these near-duplicates, in total there are 11.90 millions of distinct spammers and 741.10 millions of spam links, which constitute 4.56% and 9.50% of the total numbers of Weibo accounts and links, respectively. The largest two groups of spammers are WeiboAssistant (about 3 million spammers) and DatingGroup (about 2 millions spammers).



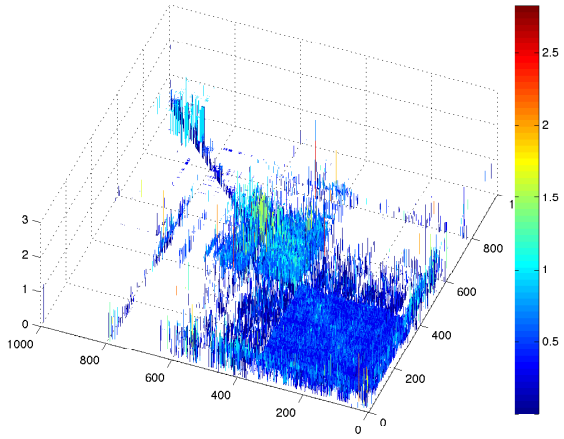
(A) Sorted by ID.



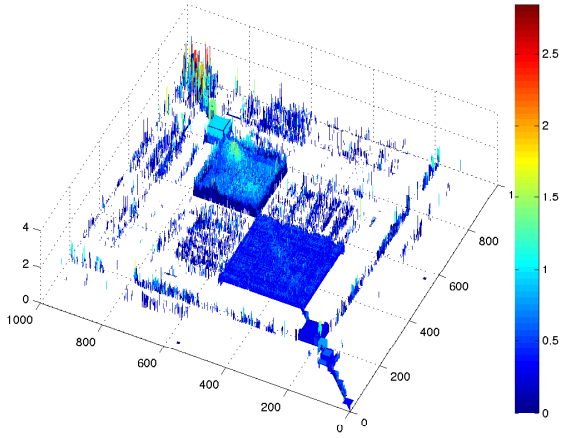
(B) Sorted by 34 clusters.

Fig. 5. Jaccard similarities between 395 near-duplicates that are sorted by IDs (A) and clusters (B).

These 395 near-duplicates and the corresponding spammers are created by a variety of spam producers. To find out their origins, we run the agglomerative hierarchical clustering algorithm on these near-duplicates, using (1-JaccardSimilarity) as the distance. Fig. 4 shows the resulting dendrogram that is created with the unweighted pair group method with arithmetic averages (UPGMA) linkage. When cutting the dendrogram at the value of 0.85, we find 34 clusters. The details of the clusters, including the statistics of the spammers and the links to the web page depicting the details of each near-duplicates, are listed in Table 1. It lists the clusters in the increasing order of the average in-degrees of the spammers. In addition to the in-degrees, we also list the average out-degrees, the number of spammers, the number of spam links, and the



(A) Sorted by ID



(B) Sorted by clusters

Fig. 6. Common target between 1000 random spammers that are sorted by IDs (A) and clusters (B). The Z-axis is in the scale of log 10.

number of near-duplicates in each cluster. For each cluster we also provide a web page that describes the details to these near-duplicates.

To verify the result of clustering, we plot the relationship between the near-duplicates and random spammers before and after the clustering in Fig. 5 and Fig. 6 respectively. For the near-duplicates, we observe that: 1) some near-duplicates IDs (e.g., the red block around IDs 250 in Panel (A) of Fig. 5) are contiguous, indicating that they are created around the same time; 2) When near-duplicates are clustered, there are small groups of near-duplicates that contain only a few members, mostly in the left lower corner in Panel (B) of Fig. 5; 3) there are several large clusters. The largest one (cluster 30) contains 117 near-duplicates, and is depicted in the upper-right corner of Fig. 5; 4) Some clusters (the first and last a few clusters) are completely

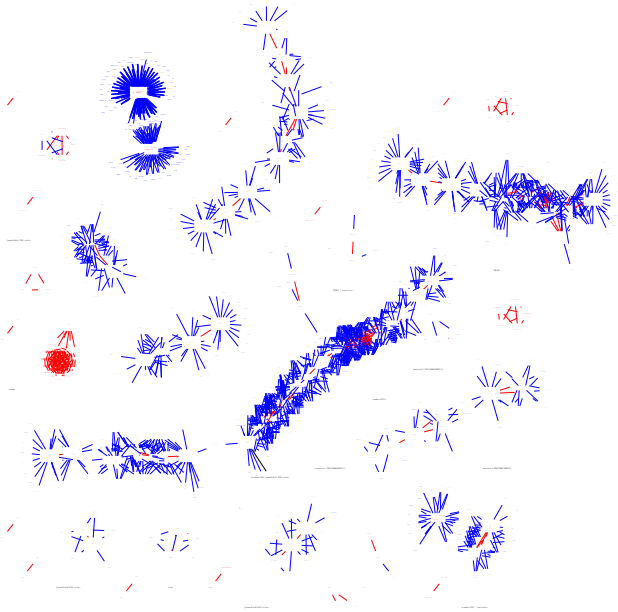


Fig. 7. The network of near-duplicates. Red nodes and edges are near-duplicate accounts and their connections. All edges originate from near-duplicates. There is an edge between a blue node B and a red node R if their similarity $S > 0.01$, where $S = F(B) \cap F(R)/min(F(B), F(R))$. A blue node is turned to orange when its *SpammedIndex* ≥ 0.5.

isolated from the remaining near-duplicates, with zero Jaccard Similarity between them. This is another evidence that they are not normal accounts. Large accounts normally have some overlapping followers.

Next, we explore the relationship between the spammers of those clusters depicted in Fig. 6. There are 12 millions of spammers, which are too large to visualize them. We select 1000 spammers uniformly at random, and plot the size of common followers between the random spammers in Fig. 6 . Panel (A) is sorted by ID, and Panel (B) by clusters. From Panel (A), we can observe that spammers are roughly grouped by their IDs, indicating different spam producers create their spam account at different time period. After clustering, we highlight the following observations: 1) there are two large groups of spammers, corresponding to WeiboAssistant (cluster 16) and DatingGroup (cluster 18). Note that these two clusters contain only five near-duplicates (two for WeiboAssistant and three for DatingGroup). Thus they are not discernible in Panel (B) of Fig. 5. Yet they have millions of spammers, thus forming two large blocks in Panel (B) of Fig. 6; 2) a large cluster of near-duplicate (cluster 30) has a small number of spammers as shown in the upper corner of panel B. Yet spammers in cluster 30 are highly integrated by sharing hundreds of common followers; 3) Since clusters

are sorted in terms of in-degrees, spammers in each cluster becomes increasingly more integrated; 4) Spammers in most clusters have the similar number of common followers.

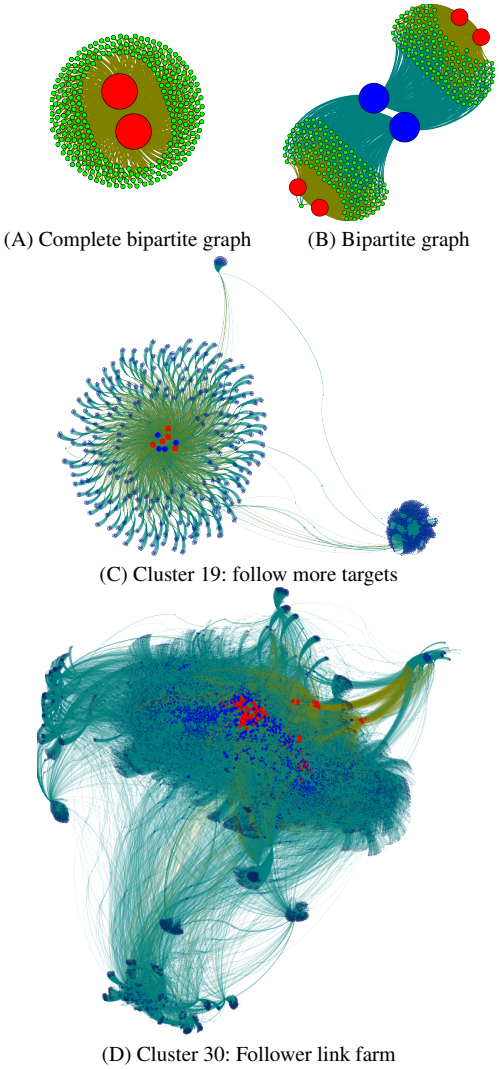
3.1 Types of Spammers

From these 34 clusters, we describe the following four representative types of spammers, ranging from simple complete bipartite graph to complex link farm. Before going into the details of these types of spammers, we summarize their properties in Fig. 10 in contrast to the random accounts in the first column.

For random accounts in Weibo, the in- and out-degrees have a heavy tail, just the same as Twitter and many other social networks. The message counts for each account also have a long tail resembling a power law. The fourth row describes the spam links vs. in-degree. For random accounts, it shows that large accounts (accounts with large in-degree) tend to receive more spam links. Row 5 depicts the location distribution, while row 6 shows the percentage of the accounts that are created in each of the 27 months. For random accounts, almost the same amount of new accounts are created during the last 10 months.

- Complete Bipartite Graph: Spammers and their targets are disjoint. The number of spam targets is very limited, and every spammer connects with every target. For example, Fig. 8 (A) is a random sample of the spammers in Cluster 1 (Love Shopping), where every spammer follows only two spam targets. Their in-degree is 0.00, out-degree is 2.00. Most probably the spammers are created for the sole purpose to boost the follower number of these two accounts. Accounts in Cluster C1 are obviously spammers as can be shown by the column two in Fig. 10: their in-degrees are mostly 0, out-degrees are two, most accounts never post any messages, and their creation time and place are also the same.
- Bipartite Graph: Spammers and their targets are disjoint. Spammers aim at more spam targets. Fig. 8 (B) illustrates the spammers in Cluster C3(Gif Animation). Every spammer follows multiple spam targets, but its out-degree is a constant (4 for these spammers).
- Power law in-degree: This kind of spammers are more sophisticated in that they try to blend in by making their in-degrees following a power law, just like most networks [15]. Spammers follow their main targets as well as some other random accounts, so that it is not obvious to detect. In contrast to the zero in-degree in the bipartite graphs, these spammers receive follow links. Interestingly, the in-degrees of spammers follow a power law. However, most of their out-degrees are the same, and their frequencies follow a log-normal distribution. Fig. 8 (C) illustrates such an example spammer group (C19, Telecom Animation) where many spammers (23%) have out-degree 28. Most of their targets are disconnected, while their main targets remain to be an obvious small set.
- Link farms: The main targets are no-longer limited to a few accounts. Spammers and their targets are closely knit–spammers are the targets of other spammers. Fig. 8 (D) illustrates a spammer cluster that involves 117 near duplicates (cluster

C30), and many other spammer targets. Spammers typically have the maximal out-degree that is allowed by the system.



(A) Complete bipartite graph (B) Bipartite graph

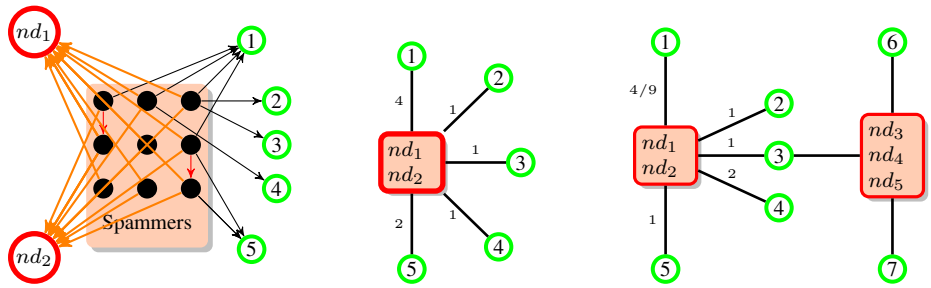
(C) Cluster 19: follow more targets

(D) Cluster 30: Follower link farm

Fig. 8. Four types of spammers.

4 Spammer Targets

To quantify the patrons of the spammers, there are two issues we need to consider: 1) A suspected spammer is not always 100% spammer. The higher is its Jaccard Similarity,



Near-duplicates and spammers (A) | Spammers reduced to one node (B) | Network of spammer groups (C)

Fig. 9. Schematic of Near-duplicate group, spam accounts, and spam links. (A) Spammers are the nodes that point to near-duplicates nd_1 and nd_2; (B) Many spammer nodes omitted, resulting in a reduced plot for the spammer group nd_1, nd_2. Nodes $1 \sim 5$ have different *Spammed Index*; (C) A spammer group can involve multiple near-duplicates(e.g. nd_3, nd_4, and nd_5). Near-duplicate groups can point to the same target (node 3), who receives *SpammedIndex* from both groups.

the higher probability it is a spammer. Therefore we introduce the *SpammerIndex* to reflect the probability of being a spammer. 2) An account may receive spammers from multiple sources. The *SpammedIndex* of an account quantifies the total number of possible spammers it receives.

4.1 Spammer Index and Spammed Index

The probability of an account being a spammer, we call it the *SpammerIndex*, is the Jaccard Similarity of the near-duplicates it created, minus the expected Jaccard similarity. More formally, we give the following definition:

Definition 2 (SpammerIndex) *Given an account i. Let a and b be the most similar accounts that both have account i as their follower. The spammer index of i is the deviation from the expected Jaccard similarity, i.e.,*

$$s_i = \begin{cases} J_{ab} - E_{ab}, & \exists ab\ s.t. J_{ab} - E_{ab} > \theta \wedge i \in F(a) \cap F(b); \\ 0, & otherwise, \end{cases} \tag{11}$$

where θ is the threshold value, E_{ab} is the expected Jaccard similarity between accounts a and b, and $F(x)$ is the set of followers of account x.

Intuitively, if node i is involved in the creation of a near-duplicate pair a and b, we say that i is a spammer with probability $J_{ab} - E_{ab}$.

Note that large accounts naturally share more followers, therefore they have higher Jaccard Similarity. Consider a hypothetical extreme case when two very large accounts includes almost all the followers. Their expected Jaccard similarity would be close to one, yet their followers should not be regarded as spammers because their expected Jaccard similarity is also close to one. Therefore, we need to deduce the expected Jaccard similarity value in the definition.

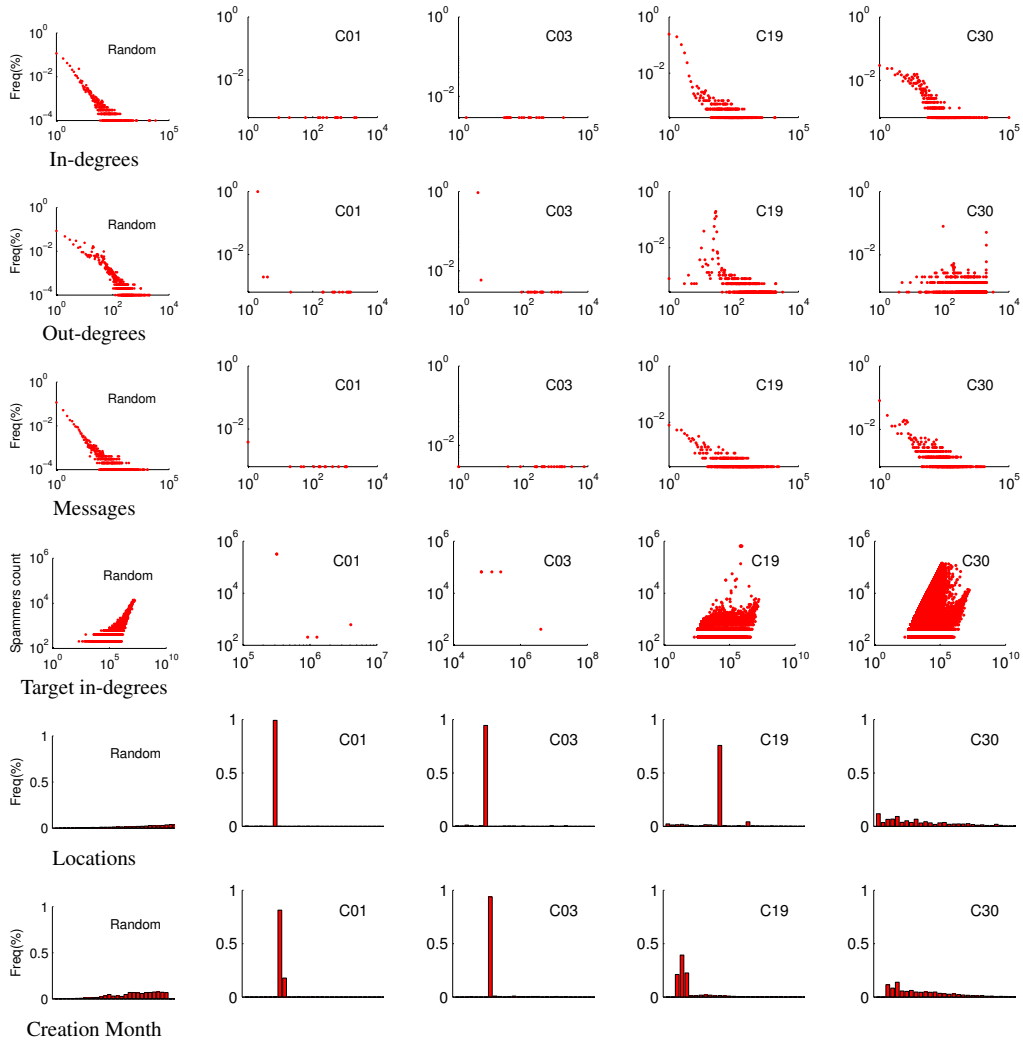


Fig. 10. Properties of four different types of clusters C1, C3, C19, and C30(columns 2 to 7), and a comparison to random accounts (column one).

Example 1 (SpammerIndex) *In Fig. 9, Suppose that there are* $N = 1000$ *number of nodes, among them 16 nodes are plotted. The remaining nodes may point to nodes* $1 \sim 5$ *only. Suppose that the normalized degree variance is* $\Gamma = 2$*. The expected number of overlapping between nodes* nd_1 *and* nd_2 *is approximately* $9 \times 9/N \times \Gamma \approx 0.16$ *[13]. The expected JS is* $0.16/9 \approx 0.02$*, assuming the degrees of both nodes* nd_1 *and* nd_2 *are 9. The SpammerIndex for each spammer node (the black dots) is* $1 - 0.02 = 0.98$*.*

Each account can have spammers as well as normal accounts as its followers. We define the *SpammedIndex* as the total amount of the spammers it contains:

Definition 3 (SpammedIndex) *The SpammedIndex S_j of an account j is the sum of the SpammerIndex it receives from its followers. I.e.,*

$$S_j = \sum_{i \in F(j)} s_i. \tag{12}$$

Intuitively, the *SpammedIndex* measures the total amount of possible spam followers. Popular accounts have large number of followers, and consequently large *SpammedIndex*. To reflect the proportion of the spammed links it receives, we define the *normalized SpammedIndex (NS)* as below:

Definition 4 (NS) *The Normalized SpammedIndex NS_j of an account j is the proportion of the spammers links it receives as followers. I.e.,*

$$NS_j = \frac{S_j}{|F(j)|} \tag{13}$$

Example 2 (SpammedIndex) *Continuing the previous example In Fig. 9, the SpammedIndex for nd_1 and nd_2 is 0.98. The SpammedIndex for node 1 is $4 \times 0.99/d_1$, where d_1 is the in-degree of node 1 that is greater than four. Note that node 1 to 8 may have incoming links not plotted.*

In sub-Fig C, node 3 receives spam index from two spam groups.

Example 3 (SpammerIndex) *Account 1002158795 follows near-duplicates DatingForMan and DatingForWoman. The Jaccard similarity between these two accounts is 0.969, while the expected Jaccard similarity is 0.053. Hence the SpammerIndex of account 1002158795 is $0.969 - 0.053 = 0.916$.*

4.2 Top Spammed Accounts in Weibo

Table 2 lists the top 20 accounts that have the highest SpammedIndex. We can see that most of them receive around one million spammers. Although their SpammedIndex is large, their spam ratio is not very high in general, because many of them have tens of millions of normal followers.

Fig. 11 depicts the relationship between the (normalized) SpammedIndex and the In-degree of the top 10,000 accounts. We find that, unsurprisingly, large accounts normally attracts more spam links, as indicated by Panel (A). It is a log-log scatter plot of the SpammedIndex as a function of the in-degree. Panel (B) is the corresponding smoothed plot with window size 100.

What is interesting is the NS, the normalized SpammedIndex. Surprisingly, we find that the value of NS for most accounts lies in two extremes, close to either zero or one, as depicted in Panel (C). This indicates that there is a large number of accounts whose followers are mostly spammers. Among the top 100,000 accounts, we find that there are 2,542 accounts whose NS are greater than 0.9, and 5448 accounts whose $NS \geq 0.5$. In other words, among the top 100,000 accounts, 5% of them are made of mostly fake-followers. Note that this number is much larger than 395 (near-duplicates), indicating that thousands of accounts receive spam links from multiple sources.

ID	Name	Followers ($\times 10^6$)	SpammedIndex ($\times 10^6$)	Normalized SpammedIndex
1642909335	微博小秘书	17.43	2.11	0.12
1654164742	微博名人	6.46	1.77	0.27
1380274560	易建联	5.87	1.24	0.21
1362607654	黄健翔	7.83	1.15	0.15
1656809190	赵薇	11.70	0.96	0.08
1197161814	李开复	9.87	0.94	0.09
1266321801	姚晨	15.42	0.91	0.06
1761047370	大嘴韩乔生	3.99	0.91	0.23
1087770692	陈坤	7.74	0.91	0.12
1182389073	任志强	5.41	0.89	0.16
1182391231	潘石屹	7.75	0.88	0.11
1682352065	周立波	9.64	0.80	0.08
1658688240	手机微博	3.02	0.80	0.26
1686326292	梁咏琪	5.94	0.79	0.13
1670071920	史玉柱	4.44	0.76	0.17
1222713954	陈志武	3.32	0.75	0.22
1470110647	于嘉	3.67	0.74	0.20
1192515960	李冰冰	8.85	0.69	0.08
1282005885	蔡康永	12.29	0.68	0.06
1650569064	朱骏	3.32	0.68	0.20

Table 2. Top 20 'polluted' accounts sorted by SpammedIndex. Near-duplicate accounts are not included.

Panel (D) is the smoothed plot with window size 100 that corresponds to Panel (C). It shows that: 1) smaller accounts are more spammed than large accounts in average; 2) the average NS fluctuates widely around 0.2 for smaller accounts, indicating that NS values are dichotomized.

4.3 The Dating group

One particular interesting group of near-duplicates is the dating group [1]. It is very large, containing 2.5 million spammers. Most spammers are not obvious ones like the AntiqueShop group. Many top bloggers have these spammers in large amount, in hundreds of thousands.

The near duplicates Three accounts, Dating, DatingForWomen, and DatingForMen are the near duplicates that share 2.5 million followers. These three accounts are no longer active at the time of writing this paper. The last post from DatingForWoman is on December 21, 2012. In total it has 723 posts, 51 friends. DatingForMan has been dormant since September 3, 2012. After that there was only one post on January 26, 2014. In total it has 3156 posts, 41 friends. The Dating account also stopped posting on March 15, 2012. It posted 1782 times. The Jaccard similarities between them are: J(Dating,Women)=0.95, J(Dating,Men)=0.96, and J(Men,Women)=0.97.

A person can be interested in looking for a date with a woman or a man, but normally not for both. The uncanny strong correlation between these three groups of followers in the amount of millions is an indication that these accounts are manipulated

[1] http://569.asxzy.net/view_node-1787709495

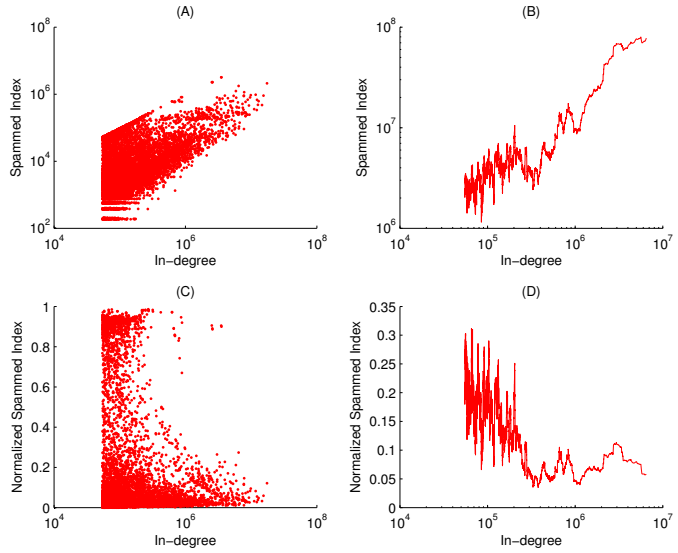


Fig. 11. Panels A and B: SpammedIndex against in-degrees. Panels C and D: Normalized SpammedIndex against in-degree. Panel B and D are obtained by smoothing with window size 100.

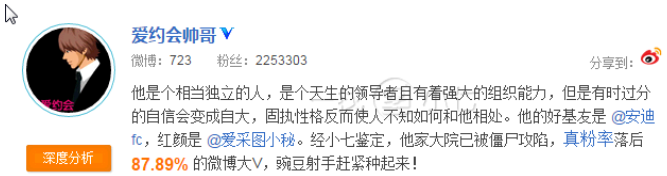


Fig. 12. Screen shot from Weibo Analytics web site `http://7.yeezhao.com/`. Fake follower rate of the Dating account is above 87.89%. Taken at the time of writing this paper.

and possibly spammers. A Weibo analytics web site also confirmed that most of them are spammers, as shown in Fig. 12.

Whom do the spammers point to Next, we study the spam targets, the accounts the spammers point to. These spammers have 1.7×10^7 out-links, account for about 2 % of the total links in the Weibo user network at the time the sample was taken. Large accounts have high visibility. Naturally, they are the major spam targets. Fig. 13 shows that the node size has positive correlation with the spam links received. What is surprising is that there are two stratified groups of spam targets. One group receives spam links in hundreds of thousands, and most of them receive the links in the same amount, around 200,000. Among this group, some accounts moved forward to attract other links. Others lag behind, having spammers as the major source of their followers. In addition, there is a large group at the left-lower corner whose number of spam links are close to their in-degrees, indicating that most of the followers are spammers from the dating group.

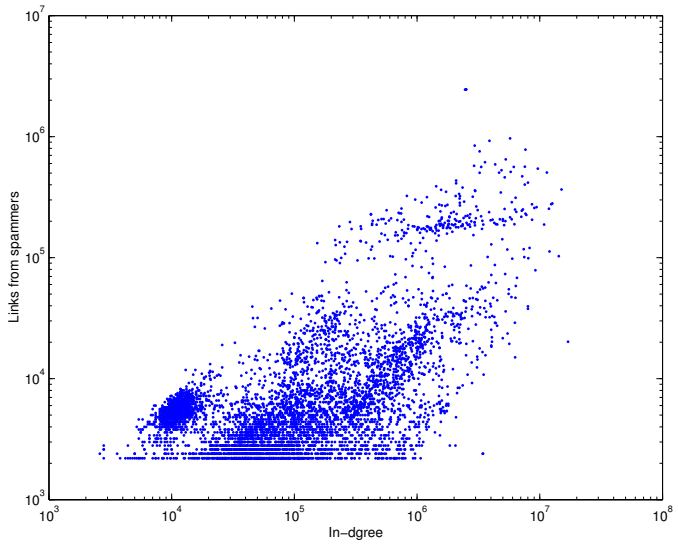


Fig. 13. Number of spam links from the Dating group as a function of the in-degree.

Who follow spammers There are very few accounts that follow these spammers. Unlike link farms in the Web, where spam pages normally point to each other to boost their PageRank values, this group of spammers aim at the simple inflation of follower number only. Many of the spammers have 0 to 3 in-links (40%).

5 Related Work

It is a challenging task to detect spammers. Two of the approaches are 1) use suspended account list given by the service provider; 2) manually label the spammers, then learn a classifier from such training data [12]. For Twitter, Ghosh et al. [5] identified 41k spammers based on two criteria: 1) they are suspended by Twitter; 2) they have posted URLs that are blacklisted by two of the most popular URL shortening services. Then they summarize the properties of these spammers collectively without clustering. Compared with this work, we identify millions of spammers, and characterize the distinctive properties and structure for spammers in each cluster.

Most spammer detection systems have some spammer accounts labeled first. Then they try to learn the characteristics of spammers. For instance in [1] 355 spammers and 710 non-spammers are used to train the model for spam detection. [9] uses both the network structure and micro-blogs to detect spammers in Twitter. It also utilizes the information of the users who are suspended by Twitter, and regard this set of users as spammers. [14] detects spammers by clustering tweets streams. [17] use honey pot to attract spammers in Facebook, MySpace and Twitter. They also train a Random Forest algorithm to find more spammers in these three OSNs.

Near-duplicates are normally used for web documents [8]. Recently it is also used to analyze Tweets in Twitters [18] [22]. These approaches study the similarity between

the documents/tweets, while we study the similarity of accounts by comparing their followers.

6 Conclusions

This paper proposes to use near-duplicates to identify spammers in OSNs. The method is conceptually simple in that it depends on the user network only, instead of individual user behaviours. The implementation is based on the estimation of Jaccard Similarity using random sampling. Unlike traditional fast algorithms for Jaccard similarity, we estimate the Jaccard similarities without the access to the entire data.

The method is applied on Weibo OSN, and find millions of spammers. We corroborate our method by detailed analysis on the spammers that are found. All the spammer groups have their highly regulated properties that make them distinct from normal accounts.

We want to emphasize that Weibo, as well as other OSNs, evolve quickly over time. Every day, many spammers are deleted, and new accounts and followers are added. Since our sampling process spans over one month (in the month of November in 2011), the estimation of the Jaccard Similarity may not be very accurate due to the dynamics of the social networks.

Our method is conservative in identifying the spammers in that 1) The threshold value for near-duplicate is 0.9. When it is reduced, much more spammers are detected; 2) We can not discover spammers that do not create near-duplicates. In this case, spammers either support one target only, or they are split into small portions and sold to a small number of consumers.

Our method can be extended to other social networks. The restriction is that it requires the accounts to be sampled uniformly at random. In the year of 2011, Weibo provided a mechanism of uniform random sampling. Now that feature is disallowed. That is why we can not discover the spammers for the current Weibo, nor can we sample the current Twitter due to the severe restriction imposed by the Twitter APIs.

The rampant spamming activities revealed in this paper prompt us the urgent needs of independent research on OSNs. OSN service providers have their own agenda and may not be interested in cleaning up the spammers. We have to resort to sampling methods to dig into the data that are hidden behind these searchable interfaces.

7 Acknowledgements

This work is supported by NSERC Discovery grant. We would like to thank Hao Wang for collecting the uniform random sample of Weibo that is used in this paper, and for his participation in the calculation of Jaccard similarity on this data.

References

1. F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, volume 6, page 12, 2010.

2. C. Chen, K. Wu, V. Srinivasan, and X. Zhang. Battling the internet water army: Detection of hidden paid posters. In *The 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2013.
3. Z. Chu and et al. Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *IEEE Transactions on Dependable and Secure Computing*, 9(6):811–824, 2012.
4. A. Dasgupta, R. Kumar, and T. Sarlos. On estimating the average degree. In *Proceedings of the 23rd international conference on World wide web*, pages 795–806. International World Wide Web Conferences Steering Committee, 2014.
5. S. Ghosh, B. Viswanath, F. Kooti, N. K. Sharma, G. Korlam, F. Benevenuto, N. Ganguly, and K. P. Gummadi. Understanding and combating link farming in the twitter social network. In *Proceedings of the 21st international conference on World Wide Web*, pages 61–70. ACM, 2012.
6. J. Giles. Social-bots infiltrate twitter and trick human users. *New Scientist*, 209(2804):28, 2011.
7. M. Gjoka, M. Kurant, C. Butts, and A. Markopoulou. A walk in facebook: Uniform sampling of users in online social networks. *Arxiv preprint arXiv:0906.0060*, 2009.
8. M. Henzinger. Finding near-duplicate web pages: a large-scale evaluation of algorithms. In *SIGIR*, pages 284–291. ACM, 2006.
9. X. Hu, J. Tang, Y. Zhang, and H. Liu. Social spammer detection in microblogging. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2633–2639. AAAI Press, 2013.
10. L. Katzir, E. Liberty, and O. Somekh. Estimating sizes of social networks via biased sampling. In *WWW*, pages 597–606. ACM, 2011.
11. S.-M. Lee and A. Chao. Estimating population size via sample coverage for closed capture-recapture models. *Biometrics*, 50(1):88–97, Mar. 1994.
12. C. Lin, J. He, Y. Zhou, X. Yang, K. Chen, and L. Song. Analysis and identification of spamming behaviors in sina weibo microblog. In *Proceedings of the 7th Workshop on Social Network Mining and Analysis*, page 5. ACM, 2013.
13. J. Lu and D. Li. Bias correction in small sample from big data. *TKDE, IEEE Transactions on Knowledge and Data Engineering*, 25(11):2658–2663, 2013.
14. Z. Miller, B. Dickinson, W. Deitrick, W. Hu, and A. H. Wang. Twitter spammer detection using data stream clustering. *Information Sciences*, 260:64–73, 2014.
15. M. Newman. *Networks: an introduction*. Oxford University Press, Inc., 2010.
16. N. Perlroth. Fake twitter followers become multimillion-dollar business. *NewYork Times*, 2013.
17. G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference on - ACSAC '10*, page 1, New York, New York, USA, Dec. 2010. ACM Press.
18. K. Tao, F. Abel, C. Hauff, G.-J. Houben, and U. Gadiraju. Groundhog day: near-duplicate detection on twitter. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1273–1284. International World Wide Web Conferences Steering Committee, 2013.
19. A. Wang. Don't follow me: Spam detection in twitter. In *International Conference on Security and Cryptography (SECRYPT)*, 2009.
20. H. Wang and J. Lu. Detect inflated follower numbers in osn using star sampling. *The IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 127–133, 2013.
21. B. Wu and B. D. Davison. Identifying link farm spam pages. In *Proceedings of the 14th International World Wide Web Conference*, pages 820–829. ACM Press, 2005.
22. Q. Zhang, H. Ma, W. Qian, and A. Zhou. Duplicate detection for identifying social spam in microblogs. In *Big Data (BigData Congress), 2013 IEEE International Congress on*, pages 141–148. IEEE, 2013.