Sockpuppet Detection in Social Network via Propagation Tree

Jiacheng Li[1,2], Wei Zhou[1]✉, Jizhong Han[1], and Songlin Hu[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{lijiacheng,zhouwei,hanjizhong,husonglin}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

Abstract. Sockpuppet detection is a valuable and challenging issue in social network. Current works are continually making efforts to detect sockpuppet based on verbal, non-verbal or network-structure features. However, they do not consider the propagation characteristic and propagation structure of sockpuppet. With our observation, the propagation trees of sockpuppet and ordinary account are different. Sockpuppet' propagation tree is evidently wider and deeper than that of the ordinary one. Based on these observations, we propose a propagation-structure based method to tackle sockpuppet detection problem. The experiment on two real-world datasets of Sina Weibo demonstrates that our method is more robust and outperforms previous methods.

Keywords: Sockpuppet Detection · Propagation Tree · Social network

1 Introduction

Social networks become a preferential place for information propagation or opinions and to promote ideas [12]. The malicious accounts on social networks lead to serious risks [9]. When the malicious accounts are detected and blocked, they register some new accounts called sockpuppets to continue spreading information. Sockpuppets usually produce malicious and deceptive behavior, such as fraud [11], cyberbullying [2], hate speech [6], and rumors [8]. Therefore, sockpuppet detection is valuable and challenging research issue. We broadly define puppetmaster as an individual that manipulate more than one account.

Prior works on automatic sockpuppet detection have tended to focus on verbal [9], non-verbal [10] and network-structure[7] features. The verbal-based method identify the authorship attribution of sockpuppet [3] by extracting features that capture stylistic, grammatical, and formatting preferences of the authors on 77 groups in Wikipedia and comparing the writing style of account [9]. It assumes that sockpuppets have a similar linguistic preference, such as keywords and topic titles in online discussion forum [15]. [4] is based on byte-level n-grams which are language independent. However, smart puppetmasters would disguise by altering account profile and writing style. Thus non-verbal methods

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_44

assume that the non-verbal behavior indicates the intention of puppetmasters, [13] extracts 11 features from contribution's behavior of the accounts, and applies the community detection algorithm to detect sockpuppet group based on the action graph and relationship graph. But most non-verbal features are not fit for different platforms. Existing network structure-based detection methods are subjectively based on user views or emotional similarities. Bu et al. [1] proposed a sockpuppet detection algorithm based on authorship-identification techniques and relationship analysis. The relationships between two accounts are built if they have a similar attitude and similar writing styles. Besides, Kumar et al. [5] constructs the reply network on discussion community and observes that the nodes denoting sockpuppets were more central and highly active. Some community detection based methods have been proposed to leverage the network structure to detect sockpuppet. However, these existing methods almost ignore the propagation characteristic and structure.

In this work, we observe that the differences of propagation trees between sockpuppet and ordinary account which are unusual patterns ignored in previous works. Sockpuppet propagation tree contains more identical accounts and is unexpectedly wider and deeper than that of the ordinary ones. In addition, the sockpuppet tend to build similar propagation trees. To utilize these patterns of the observations, we construct the propagation tree to detect sockpuppet and extract a set of independent features from propagation tree to detect sockpuppet. To validate the effectiveness, we collect two real-world data sets from Sina Weibo[1]. The experiment demonstrates that our method outperforms previous methods.

2 Problem Formulation

Suppose $G = (V, E)$ be a social network, where V is a set of accounts, $E \in V \times V$ is a set of repost relationship, and $e^i_{vu} \in E$ denotes repost relationship of message i between account v and $u (v, u \in V)$ which reflects propagation of information over G. We formally define the sockpuppet detection problem as: given a set of accounts $U (U \subset V)$, it aims to classify account $u_i (u_i \in U)$ as a sockpuppet account or ordinary account.

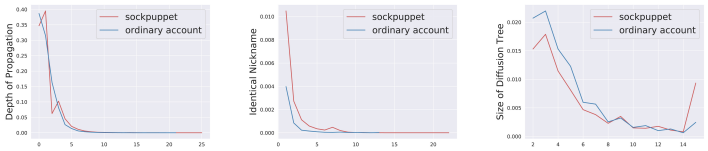
3 Observations

We engage in investigation of the difference sockpuppet and ordinary account. (1) Difference between sockpuppet and ordinary account. How difference between sockpuppet and ordinary account on dimensions of propagation tree? The number of identical nicknames. (2) The difference of pairwise accounts. Are the propagation behavior of two individual sockpuppets in the same sockpuppets group more similar than sockpuppet-ordinary account pair?

[1] https://weibo.com/

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_44

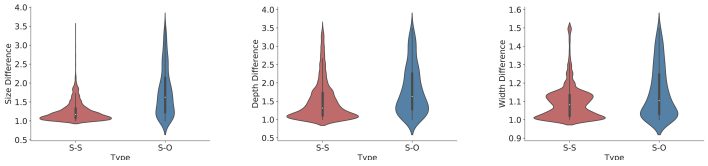
Difference between sockpuppet and ordinary account. Combined with Fig. 1b and Fig. 1c, the sockpuppet tend to participate in same discussion of post more than once, in order to maximize the influence of the post. According to structural character, the propagation tree of sockpuppet is deeper and highlights that the message is reposted by sockpuppet will be spread far(1.86 vs 1.75) and wider(4.15 vs 3.51).



(a) Depth of Tree (b) Accounts Diversity (c) Size of Tree

Fig. 1: (a) shows sockpuppet mainly retweets more than once and the ordinary account tend to do not repost it(2.03 vs 1.86). (b) demonstrates that sockpuppet is more active than ordinary account(4.60 vs 3.13). (c) illustrates that sockpuppet tend to participate hot discussion(6.09 vs 5.54).

Difference of pairwise accounts. Fig. 2 shows the sockpuppets pair is more similar than others through three dimensions: size, depth, and width. It is reasonable that the pairwise sockpuppets behave similarly. It indicates that it is hard for puppetmaster to disguise their identity on propagation behavior.



(a) Difference of Size (b) Difference of Depth(c) Difference of Width

Fig. 2: sockpuppets pair (S-S) refers to two individual sockpuppets that belong to same sockpuppets group, sockpuppet-ordinary account pair (S-O) refers to two accounts that are sockpuppet account and ordinary account separately.

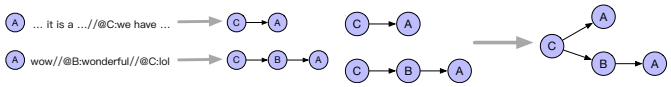
To sum up, we have several discoveries that sockpuppet tend to repost from the other sockpuppet and the message which is reposted by sockpuppet have a wider propagation range than ordinary account. The pairwise sockpuppets tend to behave similarly to each other, in order to enhance the influence of sockpuppets group opinion.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_44

4 Methodology

4.1 Propagation Tree Construction

Similar to Twitter[2], there are two types of posts in Sina Weibo: original posts (tweets) and reposts (retweets). Each reposting log will represents an information propagation process, such as "wow//!B:wonderful//@C:lol". Based on the practice of refereeing to another account in a tweet via "//@username" convention [14], we extract the usernames from reposting log and construct the propagation trees to represent the information propagation process of an account.



(a) Propagation Flow Extract (b) Propagation Tree construct

Fig. 3: (a) builds the propagation flow from reposting log.(b) constructs an propagation tree based on the same root of the propagation flow. We merge the propagation flow of account A which repost from account C. We remove the propagation tree which contains only one node.

4.2 Sockpuppet Account Detection

Given an account u and constructed the propagation trees of account u. Our method capture propagation behavior features fall into tree types: average value, minimum value and standard deviation. The average value of dimension can be seen in the following term:

Number of posts(Np_u): We count the size of set of propagation tree of account $u(D_u)$. This is a typical feature that depicts the activity frequency of accounts in social network.

Average depth of propagation tree(Ad_u): For this feature, we just count maximum depth dp_i of d_i^u. This presents the delay in the message i propagation of account u. $Ad_u = \sum_{i=0}^{Nd_u} \frac{dp_i}{Nd_u}$, where Nd_u is the size of D_u.

Average size of propagation tree(As_u): We count the total number of account (ds_i) of propagation tree of the original message i which account u latest participated(d_i^u). While this feature is trying to capture the coverage of message i which the account u is participated in: $As_u = \sum_{i=0}^{Nd_u} \frac{ds_i}{Nd_u}$

Average number of identical account in tree(Au_u): The goal of this features dn_i which is the number of the same nickname of d_i^u is to model the participation rates of account in the d_i^u. Some accounts prefer to interact with others account by reposting their posts: $Au_u = \sum_{i=0}^{Nd_u} \frac{dn_i}{Nd_u}$

[2] https://twitter.com/

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_44

Average maximum depth and width(Ad_u,Aw_u): Maximum depth dd_i is used for presenting one of dimensions of d_i^u: $Ad_u = \sum_{i=0}^{Nd_u} \frac{dd_i}{Nd_u}$. And maximum width dw_i is also used for presenting one of dimensions of d_i^u: $Aw_u = \sum_{i=0}^{Nd_u} \frac{dw_i}{Nd_u}$

Average Depth of only one 1-hop repost of original post(Ah_u): These feature present the depth dh_i of d_i^u with only one child. $Ah_u = \sum_{i=0}^{Nd_u} \frac{dh_i}{Nd_u}$

Average number of children of propagation tree(Ac_u): We take into consideration the number of children dc_i, which represents the diversity of d_i^u. We contain the propagation tree with single child: $Ac_u = \sum_{i=0}^{Nd_u} \frac{dc_i}{Nd_u}$

Average index of type of posts(Pm_u): The type of posts p_t can be divided three types with index of type: posting(1), replying(2) and reposting(3). $Pm_u = \sum_{t=0}^{Np_u} \frac{p_t}{Np_u}$

Average interval between interactions(Pi_u): This is a normalized feature where we compute the time difference between the t-th post p_t and the prior one p_{t-1}. It presents the frequency of which the account u uses the social network: $Pi_u = \sum_{i=0}^{Np_u} \frac{p_t - p_{t-1}}{Np_u}$

5 Experimental

5.1 Experimental Setup

Datasets. We conduct experiments on two real-world $\mathcal{D}_\mathcal{S}$ and $\mathcal{D}_\mathcal{T}$ which we crawled tweets from 2017.01 to 2018.10. from Sina Weibo. Accounts are identified as sockpuppets when self-reported sentence pattern such as "This is a sockpuppet of Mix" is matched or other accounts identify them as being controlled by a puppetmaster. Ordinary accounts are randomly selected from the accounts interact with sockpuppets and are not correlated to sockpuppets.

Comparison method. We consider the following baselines in sockpuppet detection. **Profile Attributes Features**: User profile is the basic information for each account, such as nickname and description. It reflects the lexical preference of puppetmaster. We employ attributes of accounts' homepage and the number of diversity of login device for sockpuppets detection problem. **Verbal Features(Verbal)** [9]: The basis of authorship attributes sockpuppets detection in Wikipedia tries to identify the sockpuppet pair by comparing writing style. It extracts 245 verbal features from each comment of account. **Non-verbal Features(Non-verbal)** [10]: It uses several variables to represent user behavior. Variables of online non-verbal behavior fall under time-independent behavior and time-dependent behavior. For all the methods, 10-fold cross validation is performed and the average results are reported.

5.2 Experimental Result And Discussion

We employ five widely used classification metrics for evaluation: precious(**P**), recall(**R**), F1-score(**F1**) and False Positive Rate(**FPR**). The Table. 1 compares several baseline methods and our proposed method over several machine learning

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_44

algorithms: Logistic regression(LR) , Support Vector Machine(SVM) , Random Forest(RF) , and Adaptive Boosting(ADA) . It shows that we obtained the best F1-score using the LR algorithm on different datasets and the LR algorithm appears the most robust among several methods.

Table 1: Sockpuppet Accounts Detection

Method	Alg	$\mathcal{D}_S$					$\mathcal{D}_T$				
		P	R	F1	ACC	FPR	P	R	F1	ACC	FPR
Profile	SVM	0.644	0.204	0.304	0.246	0.046	0.582	0.764	0.659	0.675	0.391
	RF	0.609	**0.622**	**0.608**	0.775	0.163	0.692	**0.679**	**0.681**	0.740	0.216
	LR	**0.709**	0.526	0.601	0.799	0.090	**0.711**	0.607	0.651	0.728	0.182
	ADA	0.634	0.225	0.324	0.735	0.056	0.621	0.395	0.496	0.630	0.198
Verbal	SVM	0.704	0.154	0.242	0.735	0.027	0.737	0.507	0.587	0.723	0.117
	RF	0.725	**0.578**	0.635	0.809	0.096	0.727	**0.698**	0.708	0.759	0.196
	LR	**0.804**	0.537	**0.635**	0.827	0.054	**0.781**	0.657	**0.710**	0.776	0.167
	ADA	0.735	0.241	0.347	0.750	0.042	0.727	0.545	0.612	0.724	0.144
Non-verbal	SVM	0.630	0.480	0.543	0.765	0.119	0.654	0.456	0.517	0.664	0.168
	RF	0.644	0.491	0.549	0.771	0.115	0.688	0.645	0.660	0.724	0.217
	LR	**0.781**	**0.597**	**0.674**	0.836	0.067	**0.742**	**0.662**	**0.694**	0.757	0.173
	ADA	0.474	0.061	0.103	0.713	0.021	0..575	0.264	0.356	0.618	0.122
Propagation Tree	SVM	0.792	0.511	0.618	0.820	0.054	0.743	0.571	0.637	0.734	0.147
	RF	0.771	0.598	0.663	0.828	0.078	0.746	0.657	0.693	0.760	0.165
	LR	**0.840**	**0.633**	**0.719**	0.856	0.052	**0.771**	**0.681**	**0.714**	0.771	0.163
	ADA	0.750	0.511	0.603	0.808	0.071	0.727	0.579	0.637	0.727	0.165

Due to some of the malicious sockpuppets are blocked, we cannot access their profile and some puppetmaster will apply diverse profile information in the same sockpuppets groups, the *Profile Attributes Based* method have the worst performance. *Verbal Based* method identifies sockpuppet through their linguistic traits which assume that sockpuppet have unique linguistic traits, because smart account could apply different writing style to express their idea. *Non-verbal Based* method outperform the Verbal Features method. A plausible explanation is that non-verbal cues are more powerful than verbal cues to characterize account. Our method provides better performance, which achieve the best performance in sockpuppet detection. It indicates that the propagation features based method could capture the sockpuppets' intention.

6 Conclusion

We investigate the difference between the sockpuppet and ordinary account and extract several features from the propagation tree structure to achieve the goal of sockpuppet detection. Then we evaluate the proposed methods on two real-world social network datasets over two subproblems. Compared with several methods, our model shows the best performance.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_44

References

1. Bu, Z., Xia, Z., Wang, J.: A sock puppet detection algorithm on virtual spaces. Knowledge-Based Systems **37**, 366–377 (2013)
2. Chelmis, C., Zois, D.S., Yao, M.: Mining patterns of cyberbullying on twitter. In: Data Mining Workshops (ICDMW), 2017 IEEE International Conference on. pp. 126–133. IEEE (2017)
3. Hosseinia, M., Mukherjee, A.: Detecting sockpuppets in deceptive opinion spam. In: Computational Linguistics and Intelligent Text Processing - 18th International Conference, CICLing 2017, Budapest, Hungary, April 17-23, 2017, Revised Selected Papers, Part II. pp. 255–272 (2017)
4. Kešelj, V., Peng, F., Cercone, N., Thomas, C.: N-gram-based author profiles for authorship attribution. In: Proceedings of the conference pacific association for computational linguistics, PACLING. vol. 3, pp. 255–264 (2003)
5. Kumar, S., Cheng, J., Leskovec, J., Subrahmanian, V.: An army of me: Sockpuppets in online discussion communities. In: Proceedings of the 26th International Conference on World Wide Web. pp. 857–866. International World Wide Web Conferences Steering Committee (2017)
6. Lekea, I.K., Karampelas, P.: Detecting hate speech within the terrorist argument: A greek case. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). pp. 1084–1091. IEEE (2018)
7. Liu, D., Wu, Q., Han, W., Zhou, B.: Sockpuppet gang detection on social media sites. Frontiers of Computer Science **10**(1), 124–135 (2016)
8. Ma, J., Gao, W., Wong, K.F.: Rumor detection on twitter with tree-structured recursive neural networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 1980–1989 (2018)
9. Solorio, T., Hasan, R., Mizan, M.: A case study of sockpuppet detection in wikipedia. In: Proceedings of the Workshop on Language Analysis in Social Media. pp. 59–68 (2013)
10. Tsikerdekis, M., Zeadally, S.: Multiple account identity deception detection in social media using nonverbal behavior. IEEE Transactions on Information Forensics and Security **9**(8), 1311–1321 (2014)
11. Wang, B., Gong, N.Z., Fu, H.: Gang: Detecting fraudulent users in online social networks via guilt-by-association on directed graphs. In: Data Mining (ICDM), 2017 IEEE International Conference on. pp. 465–474. IEEE (2017)
12. Yamak, Z., Saunier, J., Vercouter, L.: Detection of multiple identity manipulation in collaborative projects. In: Proceedings of the 25th International Conference Companion on World Wide Web. pp. 955–960. International World Wide Web Conferences Steering Committee (2016)
13. Yamak, Z., Saunier, J., Vercouter, L.: Sockscatch: Automatic detection and grouping of sockpuppets in social media. Knowledge-Based Systems **149**, 124–142 (2018)
14. Yang, J., Counts, S.: Predicting the speed, scale, and range of information diffusion in twitter. Icwsm **10**(2010), 355–358 (2010)
15. Zheng, X., Lai, Y.M., Chow, K.P., Hui, L.C., Yiu, S.M.: Sockpuppet detection in online discussion forums. In: Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2011 Seventh International Conference on. pp. 374–377. IEEE (2011)

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_44