GSCPM: CPM-based group spamming detection in online product reviews

Guangxia Xu[1,2], Mengxiao Hu[1], Chuang Ma[1]
[1]*School of Software Engineering, Chongqing University of Posts and Telecommunications*
[2]*Information and Communication Engineering Postdoctoral Research Station, Chongqing University*
Chongqing, China
xugx@cqupt.edu.cn, S171201001@stu.cqupt.edu.cn, machuang@cqupt.edu.cn

Mahmoud Daneshmand
School of Business
Stevens Institute of Technology
Hoboken, USA
mdaneshm@stevens.edu

***Abstract*—Online product review is becoming one of important reference indicators for people shopping, but the current product review site contains a lot of fraudulent reviews. Group review spamming, which involves a group of fraudulent reviewers writing a lot of fraudulent reviews for one or more target products, becomes the main form of review spamming. However, solutions for group spammer detection are very limited, and due to lack of ground-truth review data, this problem has never been completely solved. In this paper, we propose a novel three-step method to detect group spammers based on Clique Percolation Method (CPM) in a completely unsupervised way, called *GSCPM*. First, it utilizes clues from behavioral data (timestamp, rating) and relational data (network) to construct a suspicious reviewer graph. Then, it breaks the whole suspicious reviewer graph into k-clique clusters based on CPM, and we consider such k-clique clusters as highly suspicious candidate group spammers. Finally, it ranks candidate groups by group-based and individual-based spam indicators. We use three real-world review datasets from Yelp.com to verify the performance of our proposed method. Experimental results show that our proposed method outperforms four compared methods in terms of prediction precision.**

***Index Terms*—review spam, group spamming detection, online product review, clique percolation method**

I. Introduction

Online product review is becoming the second most trusted source of product information, second only to recommendations from family and friends, because consumers think that online product reviews reflect recommendations of "real" people [1]. However, online product reviews have become the hardest hit of fraudulent reviews. *Group review spamming*, which involves a group of fraudulent reviewers writing fraudulent reviews for one or more target products, becomes the main form of review spamming [2]. Such organized fraudulent reviewers named *group spammer*, and the group spammer is more influential and harmful than individual spammer, which has a great impact on creating a safe and reliable Internet environment. The issue of review spam has a long history. According to different research objectives, the research directions are mainly divided into the following three categories [3]: review spam detection [4], review spammer detection [5] and group spammer detection [6]. Although a number of solutions for review spam and review spammer have been proposed, studies of group spammer detection are still very limited.

Ever since Mukherjee and Liu first proposed the group spammer detection problem [7], the main solutions of group spammer detection are based on FIM (Frequent Itemsets Mining). Most of FIM-based approach includes two phases [6] [7] [8] [9]: First, using FIM technology to generate candidate group spammers, and then using ranking-based methods or machine learning methods to find real group spammers. The FIM-based approach has the following disadvantages [10]. First, the target product detected by FIM-based approach is more than two, but spam attacks on one or two products are very common in real life. Second, all reviewers in a group must review all target products in that group, which makes it easy for spammers to avoid detection. Finally, the FIM-based approach does not take into account the review time interval and the rating score deviation, so a lot of normal reviews are also likely to be misjudged as spam reviews.

Nowadays, more and more researchers use graph-based approach to detect group spammers [11] [12] [13], and some of them can overcome the shortcomings of FIM-based methods. However, due to lack of ground-truth review data, the issue of group spammer has never been completely solved. In this paper, we are trying to adopt a novel technology to detect group spammers.

Clique Percolation Method (CPM) [14], which is a method of community discovery, has properties as follows: First, nodes of the k-clique cluster generated by CPM are closely related, which coincides with the close relationship generated by the group spammer. Second, group spammers can overlap, a reviewer node can be part of several groups, which in line with the actual situation. Third, it is deterministic, for example, two runs on two review networks with the same topology will yield the same results. Choo et al. [15] have revealed that there exist implicit communities among reviewers based on reviewers behavioral features. And based on above reasons, CPM appears as a natural candidate to be used for group spammer detection. In this paper, we propose a novel three-

978-1-5386-8088-9/19/$31.00 ©2019 IEEE

step method, named *GSCPM*, which is based on CPM to generate candidate group spammers, collectively utilizes both behavioral data (timestamp, rating) and relational data (review network) to rank all candidate group spammers by spamicity. Here we summarize the contributions of this paper as follows:

1) Our proposed method *GSCPM* is a novel graph-based method which models review dataset as suspicious reviewer graph. Regard every reviewer as a node, there exists an edge between two reviewer-nodes if the two reviewers review the same product, and the intensity of the relationship between the two reviewers is determined by both the review time interval and the rating score deviation.

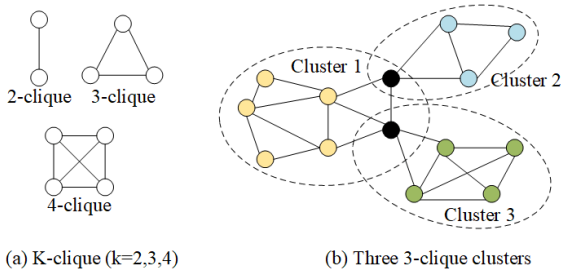
2) Based on CPM, our method breaks the whole suspicious reviewer graph into k-clique clusters. And we use k-clique cluster to describe an opinion spammer group, which are innovative in the opinion spammer group detection domain. Then rank opinion spammer groups by group-based and individual-based spam indicators, and the top ranked groups are most likely to be opinion spammer groups. Our method is completely unsupervised and does not require data annotation.

3) We use three real-world review datasets from Yelp.com to verify the performance of our proposed method. Experimental results show that our proposed method outperforms four compared methods in terms of prediction precision.

The rest of the paper is organized as follows. Section II introduces *GSCPM* in detail. Section III gives an example to show the specific steps of *GSCPM*. Section IV reports the experimental results. Finally, Section V concludes the paper and introduces some future works.

II. Proposed Method GSCPM

In this section, we propose a novel method, named *GSCPM*, which based on CPM to detect group spammers in a completely unsupervised way. CPM can find all k-clique clusters in a graph. A *k-clique* is a complete graph with k nodes. A *k-clique cluster* is a collection of adjacent k-cliques, two k-cliques are adjacent if and only if they have k-1 common nodes. Fig. 1 shows examples of k-clique and k-clique cluster. Because of the collaboration, the relationship between group spammers is closer than that between normal reviewers. Therefore, in this paper, we construct suspicious reviewer graph by review data, find all k-clique clusters in the suspicious reviewer graph, and we consider such tight k-clique clusters as highly suspicious group spammers.



(a) K-clique (k=2,3,4) (b) Three 3-clique clusters

Fig. 1. Examples of k-clique and k-clique cluster.

The overview of *GSCPM* is as shown in Fig. 2, which includes three phases. In the first phase, it constructs suspicious reviewer graph. In the second phase, it generates candidate groups based on CPM. In the third phase, it ranks candidate groups by group-based and individual-based spam indicators. Next, we will introduce the details of each phase.

A. Constructing Suspicious Reviewer Graph

Suspicious reviewer graph is built in two steps. First step, constructing reviewer graph by review data. Assuming $G = (V, E)$ is a reviewer graph, all vertex V in G represent reviewers, and there exists a edge if two reviewers have reviewed at least one common product. For $edge(i, j), i, j \in V$, the edge weight of $edge(i, j)$ represents the strength of the connection between reviewer i and reviewer j. Edge weight of "1" represents highly suspicious and edge weight of "0" represents no-suspicious. The edge weight of $edge(i, j)$ is defined as:

$$\lambda(i,j) = \begin{cases} 0, & \forall p \in P_i \cap P_j, RS(i,j,p) = 0 \\ 1, & \text{otherwise} \end{cases} \quad (1)$$

where P_i and P_j are the product sets reviewed by i and j respectively. $RS(i, j, p)$ represents the co-review similarity of i and j co-review a product p, which is defined as the review time interval and the rating score deviation. $RS(i, j, p)$ is defined as:

$$RS(i,j,p) = \begin{cases} 0, & (|t_i^p - t_j^p| > \alpha) \vee (|\gamma_i^p - \gamma_j^p| \geq 2) \\ 1, & \text{otherwise} \end{cases} \quad (2)$$

where t_i^p is the review time when reviewer i reviews product p; t_j^p is the review time when reviewer j reviews product p; γ_i^p is the rating score of reviewer i for product p; γ_j^p is the rating score of reviewer j for product p; α is a user-specified time threshold. RS of "1" represents highly suspicious and RS of "0" represents no-suspicious. Two reviewers will be judged as suspicious only if they meet both conditions of similar time and similar ratings. In the process of group spammer detection, the value of α will have a direct impact on the precision. If the α value is too large, too many real co-reviews will be included. Otherwise, it will result in missed detection.

Second step, filtering out reviewers with low suspicious. Most reviewers in the reviewer graph obtained by first step are not spammers, so we need to filter out reviewers with low suspicious. Edge weight of 0 represents no-suspicious in our method, so we ignore the edge has an edge weight of 0.

B. Generating Candidate Groups Based on CPM

In this section, it gets candidate groups based on CPM, the specific steps are as follows. First, find all k-cliques in the suspicious reviewer graph. Second, construct a clique-clique overlap matrix. The clique-clique overlap matrix is symmetric, each row (and column) represents a clique and the matrix elements are equal to the number of common nodes between the corresponding two cliques, and the diagonal entries are equal to the size of the clique. Third, construct the threshold matrix. Given a parameter k, all the off-diagonal elements that

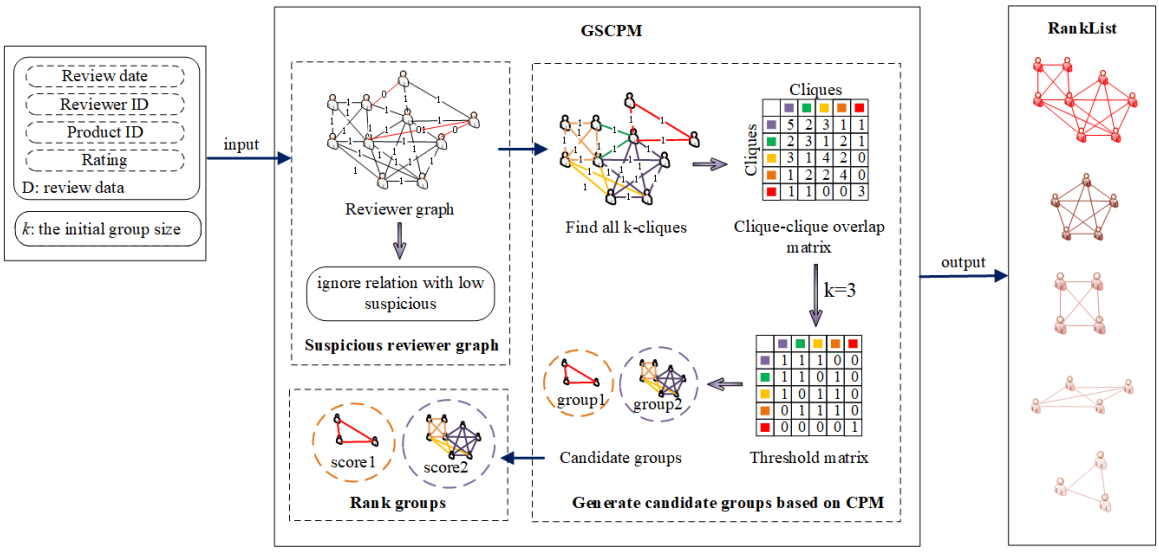


Fig. 2. Overview of GSCPM.

are less than k-1 and the diagonal elements that are less than k are set to 0, and the other elements are set to 1. Fourth, get candidate group spammers by the threshold matrix. There is a clique when the corresponding diagonal entry is 1, the two cliques are adjacent when corresponding off-diagonal element was 1.

C. Ranking Candidate Groups

In this section, it ranks candidate groups by spam score. Spam score can measure the suspicious of candidate groups, and the spam score is defined as the average of a series of spam indicators. A number of spam indicators including language indicators, behavioral indicators and joint indicators have been proposed in previous studies [16] [17] [18]. However, Xu et al. [8] find that spammers usually express themselves in a manner similar to normal reviewers, that is to say, spammers behave in the same way as normal reviewers at the level of review language. Therefore, the language indicators often perform poorly when distinguish spam/normal reviewers. Our method does not take language indicators into consideration.

In order to reduce the impact of groups with small group size, we use the following penalty function in our method:

$$L(g) = \frac{1}{1 + e^{-(|R_g| + |P_g| - 3)}} \tag{3}$$

where $|R_g|$ represents the number of reviewers in group g, $|P_g|$ represents the number of products in group g.

In our method, we measure the spamicity of a candidate group from two dimensions, individual spam indicators and group spam indicators. Three individual spam indicators are selected from [16], including:

Burstiness (BST): Spammers often review target products in a relatively short time to achieve maximum impact. Given a reviewer r, we define the time burstiness as:

$$BST(r) = \begin{cases} 0, & L(r) - F(r) > \alpha \\ 1 - \frac{L(r) - F(r)}{\alpha}, & \text{otherwise} \end{cases} \tag{4}$$

where $L(r)$ is the date of the latest review of r, $F(r)$ is the data of the first review of r, α is a user-specified time threshold, such as 10 days.

Maximum number of reviews (MNR): It is an abnormal behavior that posting a lot of reviews in one day.

$$MNR(r) = \frac{maxV_r}{max_{r \in R}(maxV_r)} \tag{5}$$

$MNR(r)$ computes the maximum number of reviews posted in a day by r, and normalize it by the maximum value of MNR.

Average rating deviation (avgRD): The average rating deviation of group g is defined as:

$$avgRD(r) = avg_{p \in P_r} \frac{\gamma_r^p - \overline{\gamma_p}}{4} \tag{6}$$

where P_r is the product set reviewed by r, γ_r^p is the rating score that r reviews product p, $\overline{\gamma_p}$ is the average rating score of product p. The maximum rating score is 5, so the maximum rating deviation is 4, and the maximum rating deviation is used for normalization.

Four group spam indicators are selected from [17] [18], including:

Review tightness (RT): The review tightness of group g is defined as:

$$RT(g) = \frac{|V_g|}{|R_g||P_g|} L(g) \tag{7}$$

where $|V_g|$ represents the number of the reviews in g, $|R_g||P_g|$ represents the cardinality of the cartesian product of the reviewer set and the product set in g. $L(g)$ is a penalty function for reducing the impact of groups with small group size.

Product tightness (PT): If a candidate opinion spammer group reviews only a few specific products and has never reviewed any other products, it is likely to be a real opinion spammer group. The product tightness of group g is defined

as:

$$PT(g) = \frac{|\bigcap_{r \in R_g} P_r|}{|\bigcup_{r \in R_g} P_r|} \tag{8}$$

Group rating deviation (GRD): The group rating deviation of group g is defined as:

$$GRD(g) = 2(1 - \frac{1}{1 + e^{-avg_{p \in P_g} S^2(g,p)}})L(g) \tag{9}$$

where $S^2(p, g)$ is the variance of the rating scores of product p by reviewers in g. $L(g)$ is a penalty function for reducing the impact of groups with small group size.

Group size (GS): The group size of group g is defined as:

$$GS(g) = \frac{1}{1 + e^{-(|R_g|-3)}} \tag{10}$$

where $|R_g|$ represents the number of reviewers in g.

The range of all spam indicators are fixed to [0, 1]. The higher the spam score of the group, the stronger the suspiciousness of the group.

III. Case Study

In this section, we give an example to show the specific steps of *GSCPM*. Assuming that the review data is as shown in Table I, the specific steps of *GSCPM* are as follows:

TABLE I
REVIEW DATA

No.	Date	Reviewer	Product	Rating
1	2012/8/09	R_1	P_1	1
2	2012/8/12	R_2	P_1	1
3	2012/8/15	R_3	P_1	2
4	2012/8/17	R_4	P_1	1
5	2012/8/13	R_9	P_2	5
6	2012/8/19	R_{10}	P_2	5
7	2012/8/17	R_5	P_2	5
8	2012/9/22	R_4	P_3	5
9	2012/9/20	R_5	P_3	5
10	2012/9/12	R_6	P_3	5
11	2012/9/18	R_7	P_3	4
12	2012/9/16	R_8	P_3	5
13	2012/11/2	R_3	P_7	5
14	2012/10/5	R_3	P_4	5
15	2012/10/10	R_4	P_4	5
16	2012/10/15	R_7	P_4	5
17	2012/10/11	R_8	P_4	5
18	2012/8/20	R_2	P_5	3
19	2012/8/13	R_5	P_5	4
20	2012/10/12	R_9	P_5	5
21	2012/8/2	R_4	P_6	5
22	2012/8/11	R_5	P_6	5
23	2012/8/3	R_6	P_6	5
24	2012/8/17	R_{10}	P_6	5
25	2012/12/1	R_5	P_8	5
26	2012/12/29	R_4	P_9	1

In the first phase, it first constructs reviewer graph of review data, the weighted reviewer graph is as shown in Fig. 3(a). second, it filters out the reviewer with low suspicious and deleting the edge has an edge weight of 0, then we get a suspicious reviewer graph, which is as shown in Fig. 3(b).

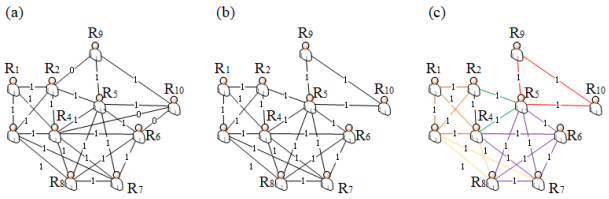


Fig. 3. Reviewer graph of the review data.

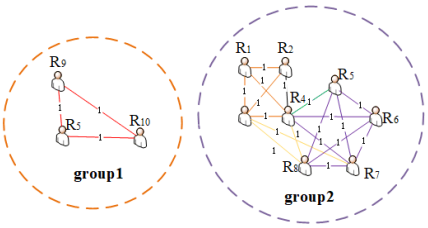


Fig. 4. Candidata spammer groups.

In the second phase, it generates candidate groups based on CPM by setting k=3. First, finding all 3-cliques as Fig. 3(c). Second, constructing clique-clique overlap matrix as A_1. Third, constructing the threshold matrix as A_2. Fourth, getting candidate groups by the threshold matrix as Fig. 4.

$$A_1 = \begin{bmatrix} 5 & 2 & 3 & 1 & 1 \\ 2 & 3 & 1 & 2 & 1 \\ 3 & 1 & 4 & 2 & 0 \\ 1 & 2 & 2 & 4 & 0 \\ 1 & 1 & 0 & 0 & 3 \end{bmatrix} A_2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

In the third phase, it computes spam score for each group. According to the spam indicators constructed in part C of Section II, we can get the spam score of 0.34 for group 1, and the spam score of 0.35 for group 2, which means that group 2 is more likely to be suspicious group spammer.

IV. Experimental Result and Analysis

A. Datasets

In this paper, we use three datasets from Yelp.com to verify the performance of our proposed method. Yelp is the largest review website in the United States, and it has a widely recognized spam filtering algorithm. Yelp does not publish details of their spam filtering algorithm, but the recommend-list and suspect-list are available on Yelp website. The three Yelp datasets used in this paper are YelpChi, YelpNYC and YelpZip, which are as shown in Table II. YelpChi [19] is the smallest dataset, which contains reviews for a set of restaurants and hotels in the Chicago area. YelpNYC and YelpZip are collected by [16]. YelpNYC contains reviews for restaurants located in NYC, and YelpZip contains restaurant reviews from a number of areas with continuous zipcode started from NYC.

TABLE II
THREE YELP DATASETS

Dataset	#Reviews (filtered %)	#Reviewers (spammer %)	#Product	Time span
YelpChi	67,395 (13.23%)	38,063 (20.33%)	201	2004.10-2012.10
YelpNYC	359,052 (10.27%)	160,225 (17.79%)	923	2004.10-2015.01
YelpZip	608,598 (13.22%)	260,277 (23.91%)	5,044	2004.10-2015.01

B. Compared Methods

We take four graph-based method as compared methods to verify the performance of *GSCPM*. The four compared methods include *GSBP*, [10] *Wang* [20], *FraudEagle* [21] and *SPEagle* [16]. It is worth noting that *Wang*, *FraudEagle* and *SPEagle* do not detect group spammers, but rank the reviews and reviewers. And *GSBP* is a group spammer detection method, which use connected graph to describe an loose spammer group.

C. Evaluation metrics

GSCPM and four compared methods are all ranking-based methods, so we use four ranking-based evaluation metrics to carry on comparative experiment. 1) Review precision @ top n: it is defined as the ratio of spam review in top n reviews; 2) Reviewer precision @ top n: it is defined as the ratio of spammers in top n reviewers; 3) Review precision @ top n groups: it is defined as the ratio of spam review in top n groups; 4) Review Recall @ top n groups: it is defined as the ratio of fake reviews in top n groups to all spam reviews; 5) Review F1-Measure @ top n groups: it considers review precision @ top n groups and review recall @ top n groups comprehensively.

D. Performance on Three Yelp Datasets

Set k=3, $\alpha = 10$ in *GSCPM*, the experimental results are as follows.

Fig. 5 shows review/reviewer precision @ top n of compared methods on YelpChi, YelpNYC and YelpZip datasets. We can see that when the review(er)s increase, the precisions of *GSCPM* and *SPEagle* are decreasing, but *GSBP*, *FraudEagle* and *Wang* weirdly go up. In general, *GSCPM* outperforms the four compared methods in terms of prediction precision, except for *SPEagle* on YelpChi dataset, which shows that *GSCPM* has better performance on larger datasets.

Fig. 6 shows review precision, recall and F1-Measure @ top n groups of *GSCPM* on YelpChi, YelpNYZ and YelpZip datasets. We can see that *GSCPM* has a better review precision on larger dataset.

E. Impact of parameters

There are two parameters in *GSCPM*, they are time threshold (α) and minimum group size (k). In this section, we investigate the impact of parameter α and k on YelpChi dataset respectively.

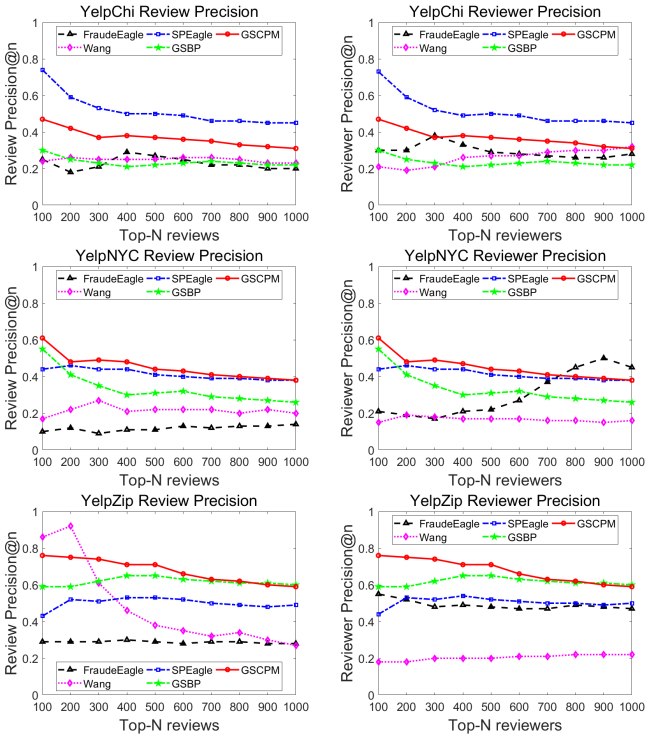


Fig. 5. Review/reviewer precision@ top n of compared method on YelpChi, YelpNYC and YelpZip datasets.

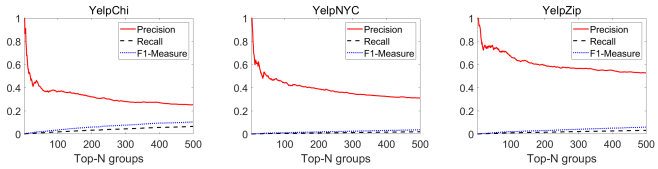


Fig. 6. Review precision, recall and F1-Measure @ top n groups of GSCPM on YelpChi, YelpNYZ and YelpZip datasets.

Fix α=10, and set k to 2, 3, 4, 5, we investigate the impact of different k on review precision, recall and F1-Measure at top n groups. From Fig. 7, we can see that as the value of k increases, review precision of *GSCPM* is gradually decreasing, while review recall and review F1-Measure are gradually increasing. In order to detect small-sized groups which include three reviewers, and consider the above experimental results, we set k=3 in this paper.

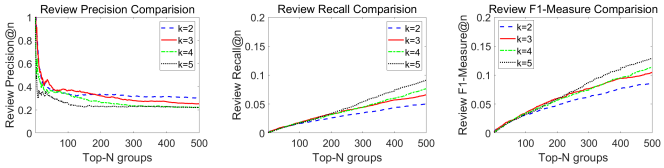


Fig. 7. The impact of minimum group size on review precision, recall and F1-Measure at top n groups.

Fix k=3, and set α to 10, 15, 20, we investigate the impact of different time threshold α on review precision, recall and F1-Measure at top n groups. From Fig. 8, we can see that the

precision, recall and F1-Measure have a little change while the value of α increases. However, as the α value increases, more reviews will be included, and computational overhead will be increased, therefore, we recommend set $\alpha = 10$ in this paper.

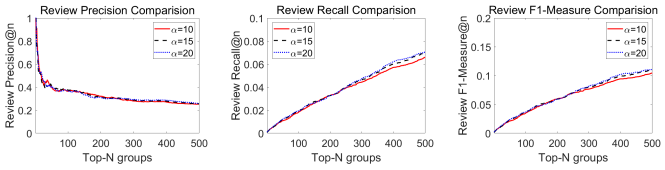


Fig. 8. The impact of different time threshold on review precision, recall and F1-Measure@ top n groups.

V. Conclusion and Future Works

In this paper, we propose a novel three-step method to detect spammer groups based on CPM in a completely unsupervised way, named *GSCPM*. First, suspicious reviewer graph of review data is constructed by behavioral data (timestamp, rating) and relational data (network). Then, *GSCPM* breaks the whole suspicious reviewer graph into k-clique clusters based on CPM, and we consider such k-clique clusters as highly suspicious candidate groups. Finally, group-based and individual-based spam indicators are defined to rank candidate groups. Three real-world review datasets from Yelp.com are used to verify the performance of our proposed method. Experimental results show that our proposed method outperforms four compared methods in terms of prediction precision. Because of lacking labeled dataset, and the overhead of manually label is huge, so we take a completely unsupervised way to detect group spammers in this paper. However, it is advisable to label a small amount of data to improve the precision of detection. Future direction is directed towards the use of partially supervised techniques.

Acknowledgment

This work was supported in part by the National Natural Science Foundation (Grant No. 61772099); the Program for Innovation Team Building at Institutions of Higher Education in Chongqing (Grant No.CXTDG201602010); the University Outstanding Achievements Transformation Funding Project of Chongqing (Grant No. KJZH17116); the Artificial Intelligence Technology Innovation Important Subject Projects of Chongqing (cstc2017rgzn-zdyf0140); The Innovation and Entrepreneurship Demonstration Team Cultivation Plan of Chongqing (cstc2017kjrc-cxcytd0063); the China Postdoctoral Fund (Grant No.2014M562282); the Project Postdoctoral Supported in Chongqing (Grant No. Xm2014039); the Wenfeng Leading Top Talent Project in CQUPT; the New Research Area Development Programme (Grant No. A2015-44); the Science and Technology Research Project of Chongqing Municipal Education Committee (Grant No. KJ1400422, Grant No. KJ1500441, Grant No. KJ1704089, Grant No. KJ1704081); the Chongqing Research Program of Basic Research and Frontier Technology (Grant No. cstc2017jcyjAX0270, Grant No. cstc2018jcyjA0672, Grant No. cstc2017jcyjAX0071); the Industry Important Subject Projects of Chongqing(Grant No. CSTC2018JSZX - CYZTZX0178, Grant No. CSTC2018JSZX - CYZTZX0185).

References

[1] F. Zhu and X. Zhang, "Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics," J. Marketing, vol. 74, no. 2, pp. 133–148, 2010.

[2] K. C. Santosh and A. Mukherjee, "On the temporal dynamics of opinion spamming: Case studies on yelp," in Proc. 25th Int. Conf. World Wide Web, 2016, pp. 369–379.

[3] A. Heydari, M. A. Tavakoli, N. Salim, and Z. Heydari, "Detection of review spam: A survey," Expert Syst. Appl., vol. 42, no.7, pp. 3634–3642, 2015.

[4] E. F. Cardoso, R. M. Silva, and T. A. Almeida, "Towards automatic filtering of fake reviews," Neurocomputing, vol. 309, pp. 106–116, 2018.

[5] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh, "Exploiting Burstiness in Reviews for Review Spammer Detection," in Proc. 7th Int. AAAI Conf. Weblogs and Social Media, 2013, pp.175–184.

[6] A. Mukherjee, B. Liu, and N. Glance, "Spotting fake reviewer groups in consumer reviews" in Proc. 21th Int. Conf. World Wide Web, 2012, pp. 191–200.

[7] A. Mukherjee, B. Liu, J. Wang, N. Glance, and N. Jindal, "Detecting group review spam," in Proc. 20th Int. Conf. World Wide Web, 2011, pp.93–94.

[8] C. Xu, J. Zhang, K. Chang, and C. Long, "Uncovering collusive spammers in Chinese review websites," in Proc. 22th ACM Int. Conf. Information and Knowledge Management, 2013, pp. 979–988.

[9] C. Xu and J. Zhang, "Towards Collusive Fraud Detection in Online Reviews," in Proc. IEEE Int. Conf. Data Mining (ICDM), 2015, pp. 1051–1056.

[10] Z. Wang, T. Hou, D. Song, Z. Li, and T. Kong, "Detecting Review Spammer Groups via Bipartite Graph Projection," Computer Journal, vol. 59, no. 6, pp. 861–874, 2016.

[11] J. Ye and L. Akoglu, "Discovering Opinion Spammer Groups by Network Footprints," in Proc. European Conf. Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), 2015, pp. 267–282.

[12] Q. N. T. Do, G. Wang, A. Zhilin, F. K. Hussain, and C. Z. P. Junior, "A network-based approach to detect spammer groups," in Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN), 2016, pp. 3642–3648.

[13] E. Choo, T. Yu, and M. Chi, "Detecting Opinion Spammer Groups Through Community Discovery and Sentiment Analysis," in Proc. Data and Applications Security and Privacy, 2015, pp. 170–187.

[14] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," Nature, vol. 435, no. 7043, pp. 814–818, 2005.

[15] E. Choo, T. Yu, M. Chi, and Y. Sun, "Revealing and incorporating implicit communities to improve recommender systems," in Proc. 15th ACM Conf. Economics and Computation, 2014, pp.489–506.

[16] S. Rayana and L. Akoglu, "Collective opinion spam detection: Bridging review networks and metadata," in Proc. 21th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2015, pp. 985–994.

[17] Z. Wang, T. Hou, D. Song, Z. Li, and T. Kong, "Detecting Review Spammer Groups via Bipartite Graph Projection," Computer Journal, vol. 59, no. 6, pp. 861–874, 2016.

[18] Z. Wang, S. Gu, X. Zhao, and X. Xu, "Graph-based review spammer group detection," Knowledge and Information Systems, vol. 55, no. 3, pp. 571–597, 2018.

[19] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, "What yelp fake review filter might be doing?" in Proc. 7th Int. Conf. Weblogs and Social Media (ICWSM), 2013, pp.409–418.

[20] G. Wang, S. Xie, B. Liu, P. S. Yu, "Review Graph Based Online Store Review Spammer Detection," in Proc. IEEE Int. Conf. Data Mining (ICDM), 2011, pp.1242–1247.

[21] L. Akoglu, R. Chandy, and C. Faloutsos, "Opinion Fraud Detection in Online Reviews by Network Effects," in Proc. Int. Conf. Weblogs and Social Media (ICWSM), 2013, pp. 2–11.