Detecting Opinion Spammer Groups

Lars Stelzer
lars.stelzer@studium.uni-hamburg.de
7346178

Niklas von Boguszewski
nvboguszewski@googlemail.com
6790872

Knowledge Processing in Intelligent Systems: Practical Seminar
Knowledge Technology, WTM, Department of Informatics, University of Hamburg

***Abstract*—Since the rise of Google Maps and Yelps business review's financial influence on local businesses, business owners have started to game the system, buy positive reviews for their businesses or demote their competitors by ordering negative ones for them. This leads most importantly to the loss of confidence and integrity to the review systems by customers. To detect such harmful reviews is a difficult task. Authors of fake reviews have started to organize themselves to boost their impact on target businesses. Furthermore, they also increased their techniques of writing fake reviews to avoid getting detected. Without a major change in nowadays review systems, spam will continue to exist and such authors will try to hide but also must leave footprints by nature. This paper tries to detect such footprints of organized fake review producing groups called opinions spamming groups by applying a graph-based method called Clique Percolation Method (CPM). The CPM algorithm will build a suspicious user graph and finds k-clique clusters in it. With aggregated opinion spamming features a suspicious score is calculated and highly suspicious groups are detected as opinion spamming groups.**

Our method changed the CPM-based method in a way that we combine our found k-cliques to artificial ones to find large organized groups.

I. Introduction

Consumers' decisions are significantly influenced by product reviews. Due to the ongoing digitalization more and more consumers rely their decision making on online reviews. Online product reviews have by now become the second must trusted source of product information, followed by recommendations from family and friends. This happens due to the belief that online product reviews reflect recommendations of "real" people. [18] A rising online marketing budget has led to a larger influence and importance of online reviews in purchase and decision making. By the fact that positive/negative reviews are able to enhance/defame products organised opinion spamming started to take place. [12] Nowadays it is estimated that up to 20% of online reviews could be fake [17].

These fake reviews are not the work of alone acting individuals, but more of organised payed group spammers [5]. More often the media reports of such groups. For example in 2013 nineteen companies got fined for their practice of writing fake online reviews. These companies systematically paid freelance writers from Philippines, Bangladesh and Eastern Europe between $1 to $10 per review. [7] With the growing popularity of these platforms and the direct influence on the businesses success the practice of leaving fake reviews for its own or the competitors' business is becoming more common.[13] The *New York Times* reported in 2011 on the case of businesses hiring workers to post fake 5-star Yelp reviews on their behalfs for as little as 25 cent per review [16]. In another case Samsung got a heavy penalty for paying people to post messages online that attacked HTC, their competitor, products while praising Samsung's [4].

Yelp filters roughly 16 % of its business reviews on its own. [13] Still reportedly Yelp's spokesman says 20-25 % of the submitted reviews are "suspicious" [7].

In this paper we propose a method to determine suspicious spammer groups accomplished by three steps. All these steps are based on the freely accessible Yelp.com dataset. Here we summarize the contributions of this paper as follows:

1. Firstly we create user groups based on the review time of the written review and the rating score given by the user. With this step we are able to create groups that reviewed the same business at nearly the same time giving the same rating score. Goal is to find groups that potentially work together writing fake reviews.
2. Followed by the creation of groups we determine features that help us to determine suspicious behaviour. In this paper we use three features that give hints if a user or group acts suspicious. We are looking at behavioral data like review time, rating score and textual similarity between reviews.
3. Based on our groups and features we are giving each group a score that determines how likely this group is an actual spammer group, called suspicious score. The suspicious score is calculated out of the key indicators given by the individual feature assigned to each group.

The paper is structured as follows. Section II introduces related work in the field of spam detection. Section III gives more detailed information about the used Yelp.com dataset, platform and review system. Section IV describes our determined spammer behavior features. Section V our model which aims to find opinion spamming groups gets described. Section VI reports the experimental results. Finally, Section

VII concludes and discusses the paper.

II. Related Work

Over the last years the field of spam review detection has been experienced plenty of attention. The existing researches can be divided based on the data and features they analyze. Below are presented three different approaches to filter and detect spam [9].

1. *Review spam detection:* this approach is based on the data of the review (e.g., textual content, meta data, etc.) to detect and filter spam reviews. [6] [11] [15]
2. *Review spammer detection:* this approach is aiming to find the spammer it self. This can be, for example, achieved by identifying atypical behavior, such as posting time or geolocation. [14] [8] [10]
3. *Group spammer detection:* goal with this approach is to identify whole group of spammers due to the fact that group spammer are more influential and harmful than individual spammer. [5] [1] [9]

Ever since Mukherjee and Liu proposed the group spammer detection problem in 2011 [2], the main approach solving this problem was using FIM (Frequent Itemset Mining). The FIM-based approach mostly included two phases: [14] [2] [3] First generating candidate group spammers using FIM technology and then using ranking-based methods or machine learning methods to find real group spammers. This approach has the following disadvantage: The generated candidate does not respresent a real spammer, all reviewers in a group must review all target products in a group and lastly review time intervals and rating score deviation does not get take in account. This disadvantages result in easy avoidance of detection by spammers and misjudgement of normal reviews as spam reviews. [9]

To avoid the above named disadvantages this paper adopted a Clique Percolation Method (CPM) by Xu, Hu and Ma [9]. This method trys to discover communities among the spammers. It was revealed by Choo et al. [5] that there exist communities between reviewers based on the behavioral features of the reviewer. By generating k-clique cluster we create relationships among the groups and spam reviewers are able to be in multiple groups at the same time.

In the upcoming sections we will explain our approach step by step. Starting with creation of groups, applying the behavioral features onto the groups and lastly with the interpretation and discussion of the results.

III. Yelp Dataset and Review System

The dataset we use is publically available on Yelp.com and consists of real-world reviews about different businesses (e.g., restaurants, hotels, etc.). Yelp is a recommendations platform based on user-generated reviews. The dataset contains information about reviews, businesses, user data, and geographical data. More than 8 million reviews and more than 200.000 businesses are included in the dataset. For this paper, we look at reviews of 2016 that appeared in the area of Charlotte in North Carolina USA. In total, we examine 51261 reviews and 5954 businesses.

IV. Spammer Behavior Features

In the field of spam, detection features can be divided into reviewer-centric and review-centric features. Review-centric features are built upon the raw text review itself and perceive what is written while reviewer-centric features are built upon metadata and describe the behavior of the reviewer within the given review system. We do use 3 features, Burstiness (BST), Extreme Rating (EXT), and Cosine Similarity (CS) whereas the BST and the EXT are reviewer-centric and the CS is review-centric.

Burstiness (BST): This feature reflects the behavior of spammers who are only active in a short timeframe with a given account. Within that timeframe, they try to write multiple reviews to maximize their impact.

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

The BST is described in [9] where $L(r)$ is the last review of a reviewer and $F(r)$ is the first one. We choose a threshold of $t = 28$ which refers to 28 days. The BST value of the group is the mean value of all given BST values of each unique reviewer in that group.

Extreme Rating (EXT): Given the 5-star rating scale of Yelp, spammers are likely to give extreme ratings due to the nature of impacting the general opinion in the given review system about the target product the most up or downwards [14].

$$EXT(r) = \begin{cases} 0 & , R(r) \in \{2, 3, 4\} \\ 1 & , R(r) \in \{1, 5\} \end{cases}$$

Given all reviews of a reviewer $R(r)$ the EXT is 1 if all of them are either 1 or 5-star rating reviews. The EXT value of a group is then calculated by the mean value of all EXT values within a certain group.

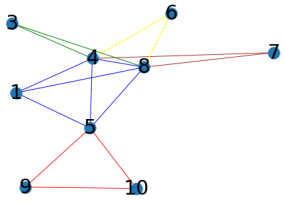
Cosine Similarity (CS): We assume that reviewers which are organized within a group of spammers are likely to post multiple duplicate/near duplicate reviews distributed among their target products. We define the CS of a group by comparing each review with one other:

$$CS(g) = \begin{cases} mean(cosine(v_j, v_k)) \\ where,\ v_j \in \{R(r_j)\},\ v_k \in \{R(r_k)\} \end{cases}$$

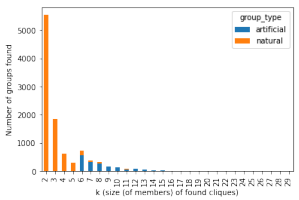
Given a group g the CS is defined as the mean value of each pairwise calculated cosine similarity of two reviews v_j and v_k among all reviews within that group.

V. Model

Our method to finding opinion spamming groups has strong similarities and was inspired by [9]. Given the Yelp dataset, we define our search space by an undirected graph $G = (U, E)$ in which U represents the users and E represents the relationship between them. To find reviewers who are acting together and are potential opinion spamming groups we take two reviewers (u_j, u_k) who posted a review to the same product, have given the same star-rating, and posted their reviews within the same 6 days time frame ($\alpha = 6$). Both users (u_j, u_k) will be added to G and build an undirected relationship e between u_j and u_k , which means that they have *acted* together. We take only tuples of reviewers (u_j, u_k) into account which have given the same star-rating duo to the fact that they either want to promote or demote the product together and don't act in opposite ways. In addition, we assume that they will act together within a certain time frame to maximize their impact on their target product. After defining the search space G we do search for k-clique clusters in it. A k-clique cluster represents a subgraph of our suspicious reviewer graph G where the clique C is defined as $C \subseteq V$, and reviewers in C (nodes) do have an undirected relationship e to each other (node). In other words, a k-clique represents a group of k-members where (at least) pairs of two members have posted a review with the same star rating and within 6 days to the same business. For our analysis we call the cliques found by this way *natural* cliques. Out of the natural cliques we build *artificial* cliques by merging pairwise 2 cliques (c_j, c_k) when both cliques share at least $\frac{2}{3}$ members ($beta = \frac{2}{3}$). This is different from the method approach used in [9].



(a) Clique examples out of the suspicious user graph G



(b) Natural and artificial group-size distribution in our dataset

An example of a set of cliques in G is shown in figure (a) where numbers represent members (nodes) and the undirected relationships are represented by the colored lines between them. Thus the members $\overline{5, 9, 10}$ build a clique with $k = 3$ analog are doing members $\overline{3, 4, 8}$, $\overline{4, 6, 8}$ and $\overline{4, 8, 7}$. The members $\overline{1, 4, 8, 5}$ build a clique where k equals 4.

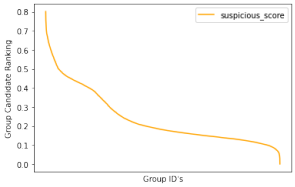
Figure (b) shows the number of members (k) in our found k-cliques divided into natural and artificial groups. The figure shows that the majority of groups are natural (8602) and their corresponding size is mostly distributed between 2 and 5 members whereas the artificial groups (1785) are starting at a group size of 6 and converging up to a group size of 29.

VI. Results/Analysis

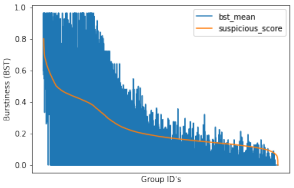
In this section, we represent the characteristics of our discovered reviewers and try to find strong arguments that separate spammers and non-spammers. Out of our 3 features (BST, EXT, CS) that do indicate opinion spamming and are applied to each group by the mean value, we calculate our suspicious score (Sus) as follows:

$$Sus = \frac{GROUP_{EXT} + GROUP_{BST} + GROUP_{CS}}{3}$$

Where $GROUP_{EXT}$, $GROUP_{BST}$, and $GROUP_{CS}$ referring to the mean value of all related values within a certain group. By dividing the sum of the features by 3 we weight all 3 features the same way. For our analysis, we normalize each feature to the corresponding max value of each feature to make it comparable to our suspicious score within one figure.



(c) Group Candidate Ranking



(d) Burstiness (BST)

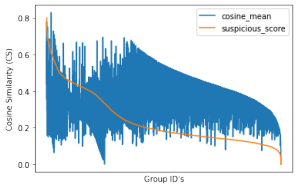
Group Candidate Ranking:

All found cliques in our suspicious user graph G are given the Sus and are sorted by it. Figure (c) shows that there are only a few groups that have a relatively high Sus. Also, table 1 shows that only 4 % of all groups do have a Sus of > 0.5. The majority of those groups are natural groups by 92 % whereas the original distribution of all groups is 82 % natural groups to 17 % of artificial ones. Is indicating that the artificial groups are not as suspicious as the natural ones in our dataset. Furthermore, the *low max* Sus value of 0.8 is indicating that to achieve a higher and more robust Sus more opinion spamming features could be taken into account.

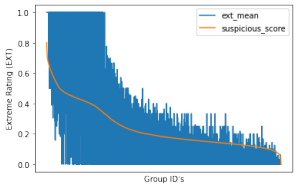
Sus_{bin}	Number of groups	Percentage
0.9 - 1.0	0	0.0%
0.8 - 0.9	1	0.0009 %
0.7 - 0.8	32	0.03 %
0.6 - 0.7	194	1 %
0.5 - 0.6	342	3 %
0.0 - 0.5	1355	13 %
0.0 - 0.5	8462	81 %

Table I: Sus_{bin} distribution by candidate groups

Burstiness (BST): Figure (d) shows that the BST value increases to almost 1 at $Sus = 0.4$ which is indicating that the top 18 % of our members in our groups are only active within a few days. Also, the top percentile of our groups does have all non 0 values which indicate the robustness of our approach.



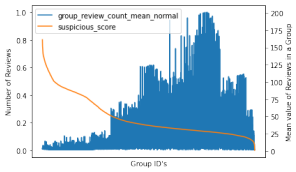
(e) Cosine Similarity (CS)



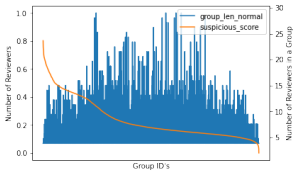
(f) Extreme Rating (EXT)

Cosine Similarity (CS): This feature is shown in figure (e) is constantly increasing up to a value of $Sus = 0.2$. Where $Sus > 0.2$ the CS value of our groups varies which indicates that some of them are duplicates/near-duplicates and others are not. We assume that this is since spammers try to avoid copy-paste nowadays to avoid getting marked/filtered as obvious spam.

Extreme Rating (EXT): The EXT in figure (f) is constantly increasing and reaches a value of 1 at $Sus = 0.4$. Although it varies between $Sus = 0.4$ and $Sus = 0.6$ the EXT value is constantly $EXT > 0$ at $Sus > 0.6$. This indicates that groups with an $EXT > 0.6$ are suspicious to demote/promote target products together.



(g) Number of Reviews



(h) Number of Reviewers

Number of Reviews: The number of reviews of each group decreases as the Sus value goes up, shown in figure (g). Where $Sus > 0.3$ the average member of a group writes 3 reviews as a $Sus < 0.3$ writes 27. We assume that this indicates that spammers only write a few reviews to reduce the damaged duo to the deletion of reviews in case they get detected. This also allows the opinion spammer to write multiple reviews for one target product.

Number of Reviewers: Figure (h) shows that the number of reviewers in each group strongly varies. We can observe that the groups at $Sus > 0.4$ are relatively small which indicates again that the majority of our highly suspicious groups are natural groups.

VII. Conclusion

This paper used an unsupervised method known as Clique Percolation Method (CPM) [9], introduced artificial groups, and used a dataset with reviews to businesses out of the year 2016 appeared in the area of Charlotte in North Carolina USA, to detect opinion spamming groups in it. The CPM-based approach shows that there are suspicious groups in our dataset that appear to be opinion spamming groups. However, in summary, we have to admit that the approach of building artificial groups out of the natural ones did not lead to a better detection of opinion spamming groups, neither lead to the detection of large organized groups with, typically more than 5 members. There are more features than our used ones to detect fake reviews which should have been also taken into account to get a higher and a more robust suspicious score. Also, our hyperparameter used to detect the groups $\alpha = 6$ and $\beta = 28$ should also be optimized on a set of different datasets to find the optimum values for them.

Besides, we could imagine calculating Sus differently or using the implemented features for a machine-learning algorithm.

References

[1] B. Liu A. Mukherjee and N. Glance. Spotting fake reviewer groups in consumer reviews. In *Proc. 21th Int. Conf. World Wide Web, 2012*, pages 191—200, 2012.

[2] J. Wang N. Glance A. Mukherjee, B. Liu and N. Jindal. Detecting group review spam. In *Proc. 20th Int. Conf. World Wide Web*, pages 93–94, 2011.

[3] K. Chang C. Xu, J. Zhang and C. Long. Uncovering collusive spammers in chinese review websites. In *Proc. 22th ACM Int. Conf. Information and Knowledge Management*, pages 979–988, 2013.

[4] JON M. CHANG. ABCNewssamsung fined for paying people to criticize htc's products, 2013.

[5] Euijin Choo, Ting Yu, and Min Chi. Detecting opinion spammer groups through community discovery and sentiment analysis. In *IFIP annual conference on data and applications security and privacy*, pages 170–187. Springer, 2015.

[6] R. M. Silva E. F. Cardoso and T. A. Almeida. Towards automatic filtering of fake reviews. In *Neurocomputing, vol. 309*, pages 106–116, 2018.

[7] Quentin Fottrell. Marketwatchyelp deems 20% of user reviews suspicious, 2013.

[8] B. Liu M. Hsu M. Castellanos G. Fei, A. Mukherjee and R. Ghosh. Exploiting burstiness in reviews for review spammer detection. In *Proc. 7th Int. AAAI Conf. Weblogs and Social Media*, pages 175–184, 2013.

[9] Chuang Ma Guangxia Xu, Mengxiao Hu. Gscpm: Cpm-based group spamming detection in online product reviews. In *IEEE International Conference on Communications (ICC)*, 2019.

[10] A. Mukherjee B. Liu J. Shao H. Li, Z. Chen. Analyzing and detecting opin- ion spam on a large-scale dataset via temporal and spatial patterns. In *Proceedings of the 9th International AAAI Conference on Web and Social Media (ICWSM)*, pages 634–637, 2015.

[11] Christopher G. Harris. Detecting deceptive opinion spam using human computation. In *Human Computation AAAI Technical Report WS-12-08*, pages 87–93, 2012.

[12] S. KC and A. Mukherjee. On the temporal dynamics of opinion spamming: Case studies on yelp. In *Proc. 25th Int. Conf. World Wide Web*, pages 369–379, 2016.

[13] M. Luca and G. Zervas. Fake it till you make it: Reputation, competition, and yelp review fraud. In *Management Science, 62(12)*, pages 3412—3427, 2016.

[14] Arjun Mukherjee, Abhinav Kumar, Bing Liu, Junhui Wang, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Spotting opinion spammers using behavioral footprints. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 632–640, 2013.

[15] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 309–319, 2011.

[16] David Segal. NYTimesa rave, a pan, or just a fake?, 2011.

[17] Wang Zhongmin. Anonymity, social image, and the competition for volunteers: A case study of the online market for reviews. In *The B.E. Journal of Economic Analysis & Policy*, pages 1–35, 2010.

[18] F. Zhu and X. Zhang. Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics. In *Journal of Marketing, vol. 74, no. 2*, pages 133–148, 2010.