Amazon Rating Data Analysis: Spam Reviewer Detection by Applying K-truss Decomposition

SHIYAN, Universiy at Buffalo
YIFAN YIN, University at Buffalo

Online shopping has become a mainstream shopping method nowadays and opinionated social media such as product reviews are now widely used by individuals and organizations for their decision making. However, due to the reason of profit or fame, people try to game the system by opinion spamming (e.g., writing fake reviews) to promote or to demote some target products. In recent years, fake review detection has attracted significant attention from both the business and research communities. However, due to the difficulty of human labeling needed for supervised learning and evaluation, the problem remains to be highly challenging.

This work proposes a graph mining angle to the problem by modeling rating relations as edges and triangles. A spam reviewer grading mechanism based on counting triangle motifs is proposed. It works on the Amazon rating dataset which provides rating records and co-bought relations between products. The intuition is that spam reviewers usually review unrelated products and give abnormal ratings. We also apply k-truss decomposition on rating data to find features of dense subgraphs helping identify behaviors of reviewers. Experiments on several Amazon rating datasets demonstrate the effectiveness of the proposed models which reflect abnormal behaviors of reviewers and automatically shows reviewers with high possibility to be spammers.

ACM Reference Format:
shiyan and Yifan Yin. 2017. Amazon Rating Data Analysis: Spam Reviewer Detection by Applying K-truss Decomposition. 1, 1 (December 2017), 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Online reviews of products and services are used extensively by consumers and businesses to make critical purchase, product design, and customer service decisions.As a recent survey has shown, 90% of consumers trust the recommendations from other consumers and 70% of customers would like to read reviews or check ratings before deciding to buy certain products[1]. It denotes that before people buy some products, they would like to look for suggestions from the ratings and reviews. However, due to the financial incentives associated with positive reviews, imposters try to game the system by posting fake reviews and giving unfair ratings to promote or demote target products and services. Such individuals are called opinion spammers and their activities are called opinion spamming. The problem of opinion spam is widespread, and many high-profile cases have been reported in the news. In fact the menace has become so serious that Yelp.com has launched a sting operation to publicly shame businesses who buy fake reviews[10]. Obviously, spam review detection is an urgent problem to be solved. Some researchers use natural language processing method to analyze fake reviews and machine learning method to predict potential spam reviews and cluster features. Some apply graph mining technology on review graphs identifying abnormal review behaviors[17]. Using classification can help scientists classify reviews then filter fake reviews.

In this paper, we established a new model to represent the relations among users and products and analyze features of these relations to detect potential spam reviewers. A graph is constructed to store relations among users and products, then use k-truss decomposition to find dense subgraphs of reviewers and products for the use of feature analysis. We also proposed a grading mechanism to display reviewer information in sorted order in terms of their scores which identify the likelihood of every reviewer to be a spam reviewer. Our method is able to find spam reviewers based on analysis on reviewers behaviors instead of the contents of reviews. We define that abnormal reviewers as those who just review products which are unrelated to each other. The count of triangles and isolated edges is used to model the relation of reviewers and products. If one user has few triangles and considerable isolated edges, to some extent, it means that this user reviews so many unrelated products that it is likely to be spammers.

To calculate these triangles, we use k-core and k-truss decomposition[2]. In addition, we use nucleus decomposition to implement the function of k-truss decomposition with more ability of extracting hierarchical features of dense subgraphs.

2 RELATED WORKS

To our best knowledge, first paper about detecting spam reviews was published by Nitin Jindal and Bing Liu [5] who used duplicate detection to find spam reviews in duplicate and near-duplicate reviews and then used machine learning model to classify spam and non-spam among the rest of reviews, called spam classification. [6] divide spam reviews into three different types. Type 1 is untruthful opinions which are deliberately faked, type 2 is reviews on brands only which are just criticize the brand manufacturers or merchants, type 3 is non-reviews which are advertisements or irrelevant contents. Type 2 and type 3 could be effectively classified, however, type 1 is hard to classify. They use duplicate and non-duplicate reviews build model to predict potential spammer. In this paper, the author describes a spam review as that praises a product is not very damaging.

Authors' addresses: shiyan, Universiy at Buffalo, Buffalo, NY, 14228, shiyan@buffalo.edu; Yifan Yin, University at Buffalo, Buffalo, NY, 14228, yifanyin@buffalo.edu.

[8] aims to find the spam reviewers in a method, which is different from traditional methods, starting from finding spam behavior instead of analyzing the reviews, and proposing ranking and supervised method to detect spammers.[7] dedicates to find the abnormal behaviors by defining different kinds of expectations and then use unexpectedness measures to rank rules.[11] aims at to detect spammer groups. They filter out candidate spam groups first, then compute values in terms of 8 criteria for groups, then rank the groups according the computed values which represents the possibility of one group to be a spam group.[17] explores the relationships among reviewers, reviews, and stores but not only reviews. It Defines trustness of reviewers, the honesty of reviews, and the reliability of stores to help judging.[15] leverages relations between the content of a review and its rating to detect potential spam reviews.[9] uses supervised methods and an unsupervised method to find spam review according to the content of reviews and behaviors of reviewers.[1] proposes a model using a multiple classifier system to detect spam reviews.

In this paper, we aims to detect spam reviewers in a behavior-based method which applies and adjusts k-truss algorithms on the graph built from the Amazon rating dataset. We use the relations between user and products to build the model.

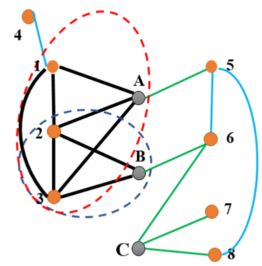

Fig. 1. Reviewer-Product Relation Model.

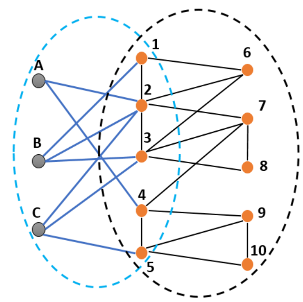

Fig. 2. Data Model.

3 MODEL AND ALGORITHM

3.1 MODEL OVERVIEW

We start by exploring some possible behavior for spamming. Grounded on common sense, we first make the following assumptions:

- Spammers are usually hired by low quality stores. Such stores have a stronger motivation to hire spammers to write dishonest reviews. Stores with good reputations and stable customer traffic may not hire spammers at all; since they lose much more if they are caught doing so. Even if good stores really entice spammers to say good things about them, it may not be very harmful. Therefore, we assume that less reliable stores are more likely to be involved in review spamming [17].
- Spammers usually review products which are unrelated since these products are not what they really need and they are required to review these products. So spammers are unlikely to review some products which have co-bought relations with those products they have reviewed. Thus, it's obvious to detect when these hired spammers behaves like this, reviewing many unrelated products in a short period. The more such behaviors performed the more likely the reviewer to be a spammer.
- Harmful spam reviews always deviate from the truth. Therefore, they can be either positive reviews about lousy stores, or negative reviews about good stores[17].

Based on these assumptions and observations, we build the following model: We extract rating relations between reviewers and products and co-bought relations among products from the Amazon rating data, then build these two kinds of relations into triangle structure which is reviewer-product-product. Figure 1 shows the model.

3.2 BUILD GRAPH

The data is the Amazon rating dataset (Reviews_Video_Games, etc.) and meta data set (Meta_Video_Games, etc.). In the rating dataset, every record consists of a reviewer id (”reviewerID”) and a product id (”asin”) which is reviewed by this reviewer and other information. In the meta data set, every record consists of a product id (”asin”), related information containing co-bought product ids (”also bought”) and other information. Relations only exist between reviewers and products, products and products. There are no relations between reviewers. Thus, reviewers and products can be constructed into a bipartite graph. The relations between reviewers and products are shown in Figure 2.

We parse these two datasets and map reviewer ids and product ids to integers and regard them as vertices in a graph. These vertices are stored in a vector where its indexes represent vertices and the values stored in this vector represent neighbors of each vertex. As Algorithm 1 states.

3.3 FINDING DENSE SUBSTRUCTURES

Finding dense substructures in a graph is a fundamental graph mining operation, with applications in bioinformatics, social networks, and visualization to name a few. Yet most standard formulations of this problem (like clique, quasi-clique, k-densest subgraph) are NP-hard[13]. Finding dense subgraphs is a critical aspect of graph mining. It has been used

ALGORITHM 1: BUILD GRAPH

Input: Amazon rating json file and metadata json file.
Output: An txt file filled with edges represented by vertices id in ascending order.

```
vertexNum = 0;
hashmap idToInt, vector graph;
while read rating data do
    reviewerID, asin = parse(line);
    put reviewerID and vertexNum to idToInt;
    vertexNum= vertexNum + 1;
    asinNum = find(asin);
    ordered_push_bask(graph[vertexNum], asinNum);
end
while read metadata do
    asinID, alsoBought = parse(line);
    put asinID and vertexNum to idToInt;
    vertexNum= vertexNum + 1;
    boughtNum = find(alsoBought);
    ordered_push_bask(graph[vertexNum], boughtNum);
end
```

ALGORITHM 2: K-TRUSS

Input: Graph G, K value.
Output: Numbers of vertices contained in k-truss.

```
reduce_to_kcore(G, k − 1);
remove_unsupported_edges(G, k);
remove_isolated_vertices(G);
return graph size;
```

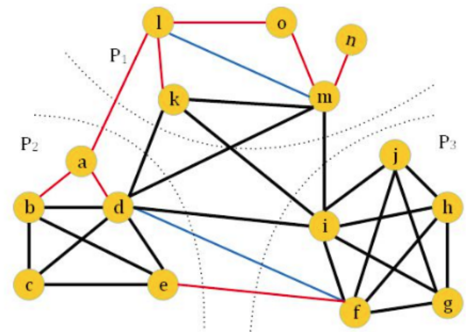

Fig. 3. The subgraph with black thick edge is a 4-truss.

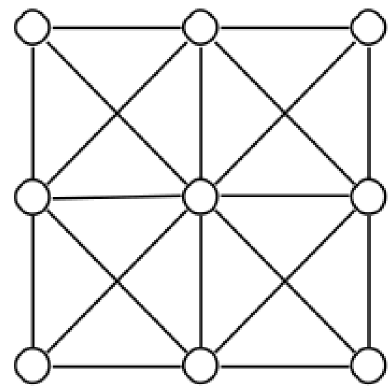

Fig. 4. 2-(2, 3) nuleus.

for finding communities and spam link farms in web graphs. It is an efficient method to reveal common features of a group of vertices in a graph. In the spam review detection problem, with our main intention that find groups of triangle relations, we use k-truss decomposition to find dense substructures in a graph and extract group features of vertices in the subgraph.

3.3.1 K-TRUSS DECOMPOSITION. k-truss decomposition is inspired by the k-core and can be thought as the same peeling problem in a higher level that deals with triangles. In 2008, Cohen et al introduced the k-truss as a better model for cohesive subgraphs in social networks, which became the most popular naming in the literature:

Definition 3.3.1 (k-truss). *[2]The truss is a non-trivial, one-component subgraph such that each edge is reinforced by at least k-2 pairs of edges making a triangle with that edge.*

corollary 3.3.2. *Each k-truss of G is a subgraph of a (k-1)-core of G.*

corollary 3.3.3. *For $k > 2$, each k-truss of G is the subgraph of a (k-1)-truss of G.*

Definition 3.3.4 (Maximal k-truss). *A maximal k-truss is a k-truss that is not a proper subgraph of another k-truss.*

We implement the k-truss decomposition algorithm and run it on the Amazon rating dataset. Given a graph $G = (V, E)$, the maximal k-truss of G can be found by removing from G those elements that are not part of the maximal k-truss. Algorithm 2 shows, first, the graph is reduced to its maximal k-1 core., This is a cheap way of dispatching many vertices and their adjacent edges before doing the harder work of eliminating edges of insufficient support. Then, remove any edge that is not supported by at least k-2 edge pairs. Finally, remove isolated vertices from G.

3.3.2 NUCLEUS DECOMPOSITION. The maximal k-truss subgraph can be obtained by the Cohen k-truss decomposition. However, it can not be used to enumerate all dense subgraphs and hierarchy structure of the input graph. So we apply nucleus decomposition on the graph.

Definition 3.3.5. *Let G be an undirected and simple graph. We use K r to denote an r-clique. Let k, r, and s be positive integers such that $r < s$. A k-(r, s)-nucleus is a maximal union $\boldsymbol{S}$ of K_ss such that:*

- *The $\boldsymbol{S}$-degree of any $R \in K_r(\boldsymbol{S})$ is at least k.*
- *Any R, $R' \in K_r(\boldsymbol{S})$ are $\boldsymbol{S}$-connected.*

For $r = 1$, $s = 2$, a k-(1,2) nucleus is a maximal (induced) connected subgraph with minimum vertex degree k[12]. This is exactly k-core[14]. Setting $r = 2$, $s = 3$ gives maximal sub-graphs where every edge participates in at least k triangles, and edges are triangle-connected. This is similar with the definition of k-truss and and k-community[16], The difference is on the connectivity condition;[16] defines k-truss and k-community as a connected component, whereas [18] do not mention connectedness, implicitly allowing disconnected subgraphs. The k-truss community defined by Huang et al. [4] is the same as the k-(2,3) nucleus: both require any pair of edges to be triangle-connected. Thus, the (2, 3) nucleus structure can be regarded as k-truss structure and can be used for finding hierarchical structure of dense subgraphs.

4 EXPERIMENTAL RESULTS

We applied our algorithms to large scale graph obtained from SNAP[3]. The vital statistics of the graph are given in Tab. 1. All the algorithms in our framework are implemented in C++. All experiments are performed on a 64-bit Ubuntu operating system running on a machine with 4 Intel Core i5-4200H 2.80GHz CPUs, with 8GB RAM.

We ran the nucleus decomposition algorithm on different Amazon rating data, containing Grocery_and_Gourmet_Food set, Automotive set, Beauty set, Cell_Phones_and_Accessories set, Clothing_Shoes_and_Jewelry set, Pet_Supplies set and Video_Games set respectively. Important statistics of these networks are given in Table 1. Vertex Number represents all reviewer and product number in the dataset, Edge Number represents all relations between reviewers and products, products and products, Rating Edge Number denotes only relations between reviewers and products, Max K denotes that the k value of the maximum (2, 3)-nucleus and the Max K containing reviewers means that the k value of the maximum (2, 3)-nucleus which contains reviewers.

The figures in Table 1 shows that the maximum k value of the whole dataset is much larger than the maximum k value of subgraphs which contain reviewers. In addition, our results show that most of reviewers are not contained in any dense subgraph. In the spam review detection, we only care about the information of reviewers, so the truss-structure dense subgraph contributes little to this problem.

In spite of this problem, we obtained some interesting group features from these dense subgraphs. We projected information of every single vertex to these dense subgraphs and obtained the result, showed in Table 2. To better analyze group features, the subgraph samples are extracted in terms of the condition: vertex number is smaller than 100 and reviewer number is larger than 2. T denotes the triangle number a reviewer vertex is involved and I denotes that the isolated rating relation number a reviewer vertex has. The result shows that most of the reviewer vertices in dense truss subgraphs are involved in more triangles structures than the number of isolated edges they have. Also, vertices in the same subgraphs usually have similar triangle number and isolated edge number.

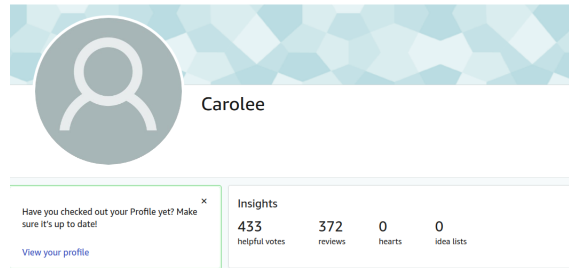

Fig. 5. Reviewer-Product Relation Model.

5 GRADING MECHANISM

As the result shows above, the method of finding dense subgraphs is good to find group features but not an effective way to specify spammers. According to previous observations, we notice that the number of products which have co-bought relation a reviewer bought and the number of unrelated products a reviewer bought can reveal some information of the reviewers behavior. Thus, we calculate the triangle number T in which a reviewer vertex v is involved and the isolated edge number I belonging to v, then calculate a score for v which equals to T I. At last, all the reviewers with a score are sorted in terms of their scores in ascending order. 7 Amazon rating data are combined to improve the accuracy. We eliminate those reviewers who has more than 10 triangles and part of the result is showed in Table 3.

According to the result, we can see that a number of reviewers have more isolated edges than triangles. Some of them attract our attention most, which have very few triangles but numerous isolated edges. The statistics show that these reviewers might have reviewerd considerable unrelated products, which make them look very suspicious. We searched the profiles of these reviewers on Amazon using their Amazon ID and analyzed their review behaviors.

The reviewer with ID A1GAS9DL6T4ZS8 has a user name Carolee, showed in Figure 5. It has 0 triangles and 199 isolated edges. After checking its profile, we found some interesting information. Figure 6 shows that this reviewer reviewed several women's pants in similar styles of different bands in two days. Then we click into the product page, showed in Figure 7, we can see that Carolee is the only one who has written a review to this product. It's a very abnormal review behavior.

The reviewer with ID A1TBBCVEO9IJ2B has a user name Reader with a shopping problem, showed in Figure 8. It has 2 triangles and 176 isolated edges. Like Carolee, this reviewer reviewed several women's tops in similar styles of different bands in one day. What's more, it gave very low ratings to most of the products it reviewed, very few high ratings. What's the same is that most of the products this reviewer has reviewed have very few rating records.

6 CONCLUSIONS

In this work, we built a reviewer-product model to represent relations among reviewers and products and help detecting

Table 1. K-truss decomposition result

Dataset	$\|V\|$	$\|E\|$	|Reviewer|	|Rating|	Max K	Max K(containing reviewers)
Grocery_and_Gourmet_Food	1060453	3350668	768438	1373768	51	8
Automotive	1364486	3337960	851418	1297156	55	8
Beauty	1685269	1685269	1210271	2023070	79	14
Cell_Phones_and_Accessories	2738290	5920711	2261045	3447249	43	4
Clothing_Shoes_and_Jewelry	5138261	26004746	3117268	5748920	61	5
Pet_Supplies	912289	2584710	740985	1235316	42	5
reviews_Video_Games	917595	2463561	826767	1324753	68	4

Table 2. Features of dense subgraphs

Dataset	|Dense Subgraph|	|T >= I Ratio = 1|	|T >= I Ratio > 0.5|	|T >= I Ratio <= 0.5|
Grocery_and_Gourmet_Food	35	22	10	3
Automotive	66	41	20	5
Beauty	63	34	17	12
Cell_Phones_and_Accessories	49	24	19	6
Clothing_Shoes_and_Jewelry	58	18	19	21
Pet_Supplies	22	11	8	3
Video_Games	22	10	9	3

Table 3. Ranking of possible spam reviewers obtained by grading mechanism (Part)

ID	|T|	|E|	Score
A1GAS9DL6T4ZS8	0	199	-199
A1TBBCVEO9IJ2B	2	176	-174
A1QUMGTMT0PWYH	2	128	-126
A1OA2ZW406NQXM	4	128	-124
A1TF098FU5DIXO	0	122	-122
A208Y1ZPM9OUNF	0	120	-120
A2DG1NQM08X1EI	10	128	-118
A345VLZZ08XXE6	0	114	-114
AX5O7O9DALUP4	0	107	-107
A1DLW4Y9QK34ZB	0	105	-105
A2JUJJLF1Y2PGR	2	107	-105
A2DQ7LI02VJMKG	2	106	-104

abnormal review behaviors effectively. We first implemented k-truss decomposition to find general information of the Amazon rating dataset, then we utilized nucleus decomposition to find hierarchy structure of the dataset and features of dense subgraphs. In this process, we found that most of reviewers were not involved in dense subgraphs so that k-truss decomposition was not an effective method to identify the feature of a single reviewer. Thus we developed a new grading mechanism to find triangles and isolated edges related to reviewers which was used to denote the possibility a reviewer could be a spammer.

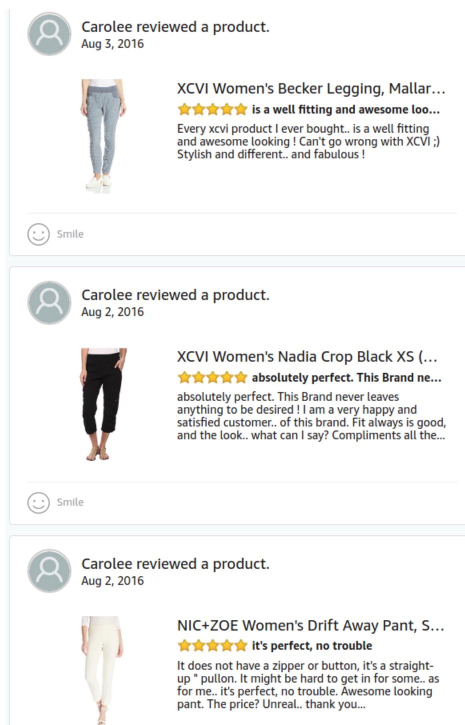

Fig. 6. Reviewer-Product Relation Model.

7 FUTURE WORK

There are two kinds of work attract us to look at in more depth. First, new effective models need to be established to

Fig. 7. Reviewer-Product Relation Model.

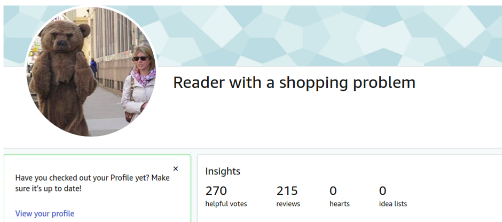

Fig. 8. Reviewer-Product Relation Model.

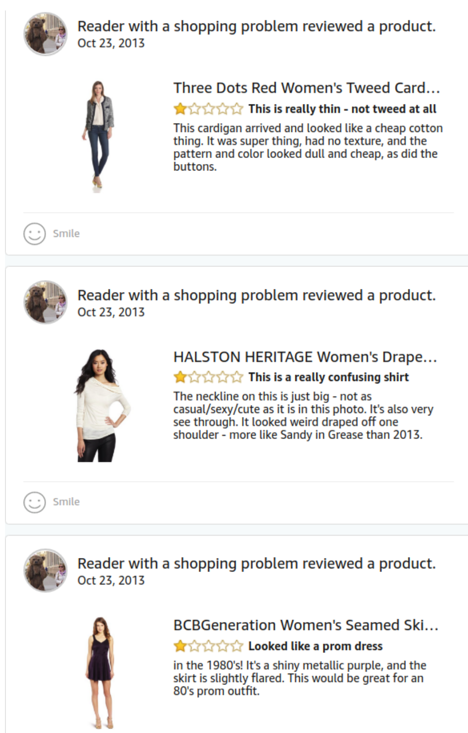

Fig. 9. Reviewer-Product Relation Model.

denote behaviors of reviewers. Second is the analysis of dense substructures. Finding group features of reviewers could be an efficient and interesting method to detect spam reviewers. Thus, more motifs are worth to try besides triangles.

Fig. 10. Reviewer-Product Relation Model.

REFERENCES

[1] Cagatay Catal and Suat Guldan. 2017. Product Review Management Software Based on Multiple Classifiers. *IET Software* (2017).

[2] Jonathan Cohen. 2008. Trusses: Cohesive subgraphs for social network analysis. *National Security Agency Technical Report* 16 (2008).

[3] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 507–517.

[4] Xin Huang, Hong Cheng, Lu Qin, Wentao Tian, and Jeffrey Xu Yu. 2014. Querying k-truss community in large and dynamic graphs. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 1311–1322.

[5] Nitin Jindal and Bing Liu. 2007. Review spam detection. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 1189–1190.

[6] Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*. ACM, 219–230.

[7] Nitin Jindal, Bing Liu, and Ee-Peng Lim. 2010. Finding unusual review patterns using unexpected rules. In *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 1549–1552.

[8] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. 2010. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 939–948.

[9] Yuming Lin, Tao Zhu, Xiaoling Wang, Jingwei Zhang, and Aoying Zhou. 2014. Towards online review spam detection. In *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 341–342.

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

[11] Arjun Mukherjee, Bing Liu, Junhui Wang, Natalie Glance, and Nitin Jindal. 2011. Detecting group review spam. In *Proceedings of the 20th international conference companion on World wide web*. ACM, 93–94.

[12] Ahmet Erdem Sariyüce and Ali Pinar. 2016. Fast hierarchy construction for dense subgraphs. *Proceedings of the VLDB Endowment* 10, 3 (2016), 97–108.

[13] Ahmet Erdem Sariyuce, C Seshadhri, Ali Pinar, and Umit V Catalyurek. 2015. Finding the hierarchy of dense subgraphs using nucleus decompositions. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 927–937.

[14] Stephen B Seidman. 1983. Network structure and minimum degree. *Social networks* 5, 3 (1983), 269–287.

[15] Kuldeep Sharma and King-Ip Lin. 2013. Review spam detector with rating consistency check. In *Proceedings of the 51st ACM Southeast Conference*. ACM, 34.

[16] Anurag Verma and Sergiy Butenko. 2012. Network clustering via clique relaxations: A community based approach. *Graph Partitioning and Graph Clustering* 588 (2012), 129.

[17] Guan Wang, Sihong Xie, Bing Liu, and S Yu Philip. 2011. Review graph based online store review spammer detection. In *Data mining (icdm), 2011 ieee 11th international conference on*. IEEE, 1242–1247.

[18] Yang Zhang and Srinivasan Parthasarathy. 2012. Extracting analyzing and visualizing triangle k-core motifs within networks. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*. IEEE, 1049–1060.