Detecting Review Spammer Groups via Bipartite Graph Projection

ZHOU WANG*, TINGTING HOU, DAWEI SONG, ZHUN LI AND TIANQI KONG

School of Information Science and Engineering, Shenyang Ligong University, Shenyang 110159, China
*Corresponding author: sylawz@foxmail.com

Online product reviews play an important role in E-commerce websites because most customers read and rely on them when making purchases. For the sake of profit or reputation, review spammers deliberately write fake reviews to promote or demote target products, some even fraudulently work in groups to try and control the sentiment about a product. To detect such spammer groups, previous work exploits frequent itemset mining (FIM) to generate candidate spammer groups, which can only find tightly coupled groups, i.e. each reviewer in the group reviews every target product. In this paper, we present the loose spammer group detection problem, i.e. each group member is not required to review every target product. We solve this problem using bipartite graph projection. We propose a set of group spam indicators to measure the spamicity of a loose spammer group, and design a novel algorithm to identify highly suspicious loose spammer groups in a divide and conquer manner. Experimental results show that our method not only can find loose spammer groups with high precision and recall, but also can generate more meaningful candidate spammer groups than FIM, thus it can also be used as an alternative preprocessing tool for existing FIM-based approaches.

Keywords: review spam; review spammer group; product review; opinion mining; bipartite graph; frequent itemset mining

Received 26 January 2015; revised 28 July 2015
Handling editor: Fionn Murtagh

1. INTRODUCTION

Online review systems are becoming increasingly prevalent nowadays. Consumers rely heavily upon consumer reviews when making decisions about which products or services to purchase online. If they read many positive reviews of a product, they might feel safe and are willing to buy it. Otherwise, they will probably refuse to buy it. Driven by the immense financial profits from product reviews, some unscrupulous individuals or organizations deliberately post untruthful reviews to promote their products or to demote their competitors’ products, trying to mislead or influence customers. Such individuals or organizations are called review spammers or fake reviewers, the reviews they write are called fake reviews, and the action is called review spamming.

To completely take control of the sentiment about a product, organizations might hire groups of reviewers to write a large number of reviews toward a set of products, which we called target products. Such review spammer groups are even more damaging since they can dramatically change the reputation of products.

Ever since Jindal and Liu first proposed the product review spamming problem in [1], much work has been done to detect fake reviews and/or review spammers. The methods can be roughly categorized into machine learning based [1–4], probability based [5, 6], behavior based [7, 8], graph based [9] and rule based [10]. Review spammer group detection, however, has not been so extensively studied. With our best effort, we found three papers which address this problem [11–13]. Since a review spam group involves a set of reviewers co-reviewing a set of products, all these works use frequent itemset mining (FIM) to find candidate spammer groups, and then build models to identify real spam groups from candidate groups. In this FIM scenario, they take reviewer identifiers as items, and products as transactions. By setting the minimum support count to 3, they can find candidate groups which have at least two reviewers and each reviewer at least reviews three common products.

However, there are many drawbacks for using FIM to generate candidate groups. (i) Due to combinatorial explosion [11], for a large dataset with a huge number of reviewers...
and products, the minimum support count used in FIM cannot be <3, which implies that only a group working on at least three products can be detected. However, groups aiming at promoting 1 or 2 products are very common in the real world. (ii) Group spammers often have to finish their tasks in a prescribed time limit, whereas the FIM-based method does not take into account the time window when generating candidate spammer groups. In FIM candidate groups, many more relevant reviewers who review the same products in a different time window might also be included by coincidence, and a reviewer can be in multiple candidate groups, so a large number of candidate groups are returned by FIM, which leads to low quality candidate groups. (iii) In each FIM candidate group, every reviewer must have reviewed all the common products reviewed by each member of the group. However, review spamming strategies are evolving. Group spammers can adjust their spamming tactics to avoid being caught by spam detecting systems. For instance, large spam groups can be divided into small sub-groups, each fulfills its own campaign [12]. In such a group, group members do not necessarily review all the target products, forming a loosely connected group, which is hard to be detected by the FIM technique.

In this paper, we use bipartite graph projection [14] to tackle loosely connected spam groups. The review data can be represented as a bipartite graph. We first construct the reviewer bipartition from the bipartite graph, and then find all the suspicious groups using a divide and conquer strategy. We claim the following contributions.

(i) We present the loose spammer group concept and its data model, which is often the case in real-world spamming activities.

(ii) We propose a set of group spam indicators derived from structural and behavioral information of reviews, reviewers and products. Human evaluation verifies the discriminating capability of these indicators in identifying spam and non-spam groups.

(iii) We model review data as a bipartite graph, and design a novel divide and conquer-based algorithm to generate loose spammer groups from the bipartite graph, which can efficiently generate high quality loosely connected review spammer groups.

(iv) We conduct a series of experiments to evaluate the effectiveness of our methods. Experiments show that our proposed method can detect spammer groups with a high precision and recall, and the algorithm is robust to predefined parameters.

The remainder of the paper is organized as follows. Section 2 discusses the related work. Section 3 formalizes the loose spammer group detection problem. Section 4 proposes the group spam indicators and the spammer group detection algorithm. Section 5 reports the experimental results of our methodology. Finally we conclude the paper in Section 6.

2. RELATED WORK

Unlike Email spam or Web page spam, which are quite straightforward to be judged by human being or computer software, online product reviews are often deceptive thus it is very difficult to judge a spam review by merely reading the review text. Therefore, classical supervised learning algorithms often fail to tackle this problem due to lacking labeled dataset.

Many researchers focus on individual fake review reviewer detection. Jindal and Liu [1] use heuristic methods to train models using features extracted from reviews, reviewers and products. They treat duplicate (or near-duplicate) reviews as fake reviews and well-chosen non-duplicate reviews as truthful reviews. In [2], a gold-standard dataset was created by hiring Amazon Mechanical Turks to write deceptive reviews of chosen hotels, while the truthful reviews were mined from real online hotels. With this dataset, detecting deceptive review tasks are issued by human judges, genre identification, psycholinguistic method and text categorization, respectively. Lim et al. [7] proposed scoring methods to measure the degree of spamicity of a reviewer based on all sorts of reviewer’s rating behaviors. Mukherjee et al. proposed an unsupervised Author Spamicity Model [5]. It works in the Bayesian setting, which facilitates modeling spamicity of authors as latent and can exploit various observed behavioral footprints of reviewers. Jindal et al. proposed a rule-based approach to find unusual review patterns by mining unexpected rules through a review dataset [10]. Wang et al. [9] proposed graph-based (or relation-based) fake reviewer detection model. Fei et al. [6] exploit burstiness in reviews to detect fake reviewers. Lappas views the fake review problem from an attacker’s perspective [8]. For reviewers who post only one or two reviews, Xie et al. [15] proposed a temporal method to address the singleton review problem. Although many approaches have been proposed to deal with the fake review/reviewer detection problem, there is no overall winner among these methods. A specific method intends to find specific spammers of certain characteristics. An idealistic approach is to combine as many methods as possible to build a multi-faceted fake review/reviewer detection system.

Mukherjee et al. [11] first proposed review spammer group detection problem. The authors first use FIM method to find candidate spammer groups, and then build a ranking model which exploits the relationships among groups, reviewers and products to compute the spamicity of a group, using features derived from the collusion phenomenon of reviewers. Xu et al. [12] also uses FIM to search for groups of reviewers who have reviewed multiple common products. They evaluated the effectiveness of frequently used spam indicators for both group-spamming and individual spamming on a large Chinese review website. However, their methods only aim to detect collusive individuals rather than spammer groups. Ref. [13] employs FIM technique to find bicliques or sub-bicliques candidates, then check them to find real collusion groups by group spam indicators. A biclique is referred to as a group of reviewers.
and a set of products, in which all the reviewers have rated all the products. A sub-biclique is a smaller sub-graph inside a biclique, which is also mined by FIM technique. Nonetheless, the paper only uses rating scores to rank collusive groups. Unlike these methods, which assume that each reviewer in the group reviews every product, we give a new definition of review spammer group, and use a different technique to mine candidate spammer groups, thus our approach generates quite different kind of spammer groups, which cannot be detected by FIM-based approaches.

3. PROBLEM DEFINITION

In this section, we propose the loose spammer group concept and its data model. We note that loose spammer groups are very common in real life, and FIM-generated candidate spammer groups are only the special cases of loose spammer groups. For ease of presentation, we list the relevant notations and their meanings in Table 1.

**Definition 3.1.** Loose spammer group: a loose spammer group (or a spammer group for short) $g$ is modeled as a quintuple form $(R_g, P_g, V_g, S_g, \tau)$, where $R_g \subseteq \mathcal{R}$ is the set of review spammers (or members) in group $g$, $P_g \subseteq \mathcal{P}$ is the set of target products in group $g$, which refers to the products that are commonly reviewed by at least two review spammers in $R_g$, $V_g \subseteq \mathcal{V}$ is the set of reviews written by reviewers in $R_g$ toward products in $P_g$, $V_g \subseteq R_g \times P_g$, $S_g$ is the set of spam indicators measuring the spamnicity of group $g$ from different dimensions, and $\tau$ is a time window which is used to choose the appropriate reviews in $V_g$ so that only the reviews occurring in group spamming time period are considered.

Given a spammer group $g$, the tightness of $g$ is defined as to what extent the group members cooperate to write fake reviews toward target products.

**Definition 3.2.** Spammer group tightness: given a spammer group $g$, the tightness of $g$, denoted by $T(g)$, is defined as the ratio of the number of reviews in $V_g$ to the cardinality of Cartesian product of $g$'s reviewer set $R_g$ and $g$'s product set $P_g$:

$$T(g) = \frac{|V_g|}{|R_g||P_g|}$$

(1)

Obviously, if each member in $g$ reviews all the target products in $g$, then $T(g) = 1$, which corresponds to the spammer groups identified by FIM-based methods as in [11–13].

**Definition 3.3.** Group spam score: given a spammer group $g$, its group spam score, denoted by $SS(g)$, ranging from 0 to 1, is a function of group spam indicators $S_g$, $SS(g) = f(S_g)$. The group spam score reflects the degree of spamming activities of $g$.

Note that there are many ways to define the group spam score function in accordance with application background. In general, it is safe to define the group spam score of group $g$ as the average value of all the spam indicators.

Given a spammer group $g$, the members in $g$ inherently form a graph structure. The node set of the graph is $R_g$, and for any two members in the group, their adjacent edge reflects the number of common target products in $P_g$.

**Definition 3.4.** Spammer group graph: given a spammer group $g$, $g$'s spammer group graph is a weighted graph, denoted by $G_g = (R_g, E)$, where $E \subseteq R_g \times R_g$ is the edge set. Given member $r_1, r_2 \in R_g$, the weight of edge $(r_1, r_2)$, denoted by $\omega(r_1, r_2)$, equals the number of common products in $P_g$ reviewed by $r_1$ and $r_2$ within time window $\tau$.

**Definition 3.5.** $k$-Connectivity spammer group: given a spammer group $g$, if all the edges in $G_g$ whose weights are less than $k$ are removed and $G_g$ remains a connected graph, then we call $g$ a $k$-connectivity spammer group, denoted by $g^k$, and its spammer group graph is $G^k_g = (R_g, E_k)$, where for any $e \in E_k$, $\omega(e) \geq k$.

The goal of review spammer group detection problem is to find all the $g^k$-s($k \geq 1$) with large $SS(g^k)$ values. For ease of expression, we will still use $g$ rather than $g^k$ to represent a $k$-connectivity spammer group if not causing confusion.

---

**Table 1.** Notation table.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r$</td>
<td>A reviewer</td>
</tr>
<tr>
<td>$v$</td>
<td>A review</td>
</tr>
<tr>
<td>$p$</td>
<td>A product</td>
</tr>
<tr>
<td>$g$</td>
<td>A spammer group</td>
</tr>
<tr>
<td>$g^k$</td>
<td>A $k$-connectivity spammer group</td>
</tr>
<tr>
<td>$\mathcal{R}$</td>
<td>Reviewer set</td>
</tr>
<tr>
<td>$\mathcal{V}$</td>
<td>Review set</td>
</tr>
<tr>
<td>$\mathcal{P}$</td>
<td>Product set</td>
</tr>
<tr>
<td>$R_g$</td>
<td>Review spammer set of group $g$</td>
</tr>
<tr>
<td>$P_g$</td>
<td>Target product set of group $g$</td>
</tr>
<tr>
<td>$V_g$</td>
<td>Spam review set of group $g$</td>
</tr>
<tr>
<td>$S_g$</td>
<td>Spam indicators of group $g$</td>
</tr>
<tr>
<td>$G_g$</td>
<td>Spammer group graph of group $g$</td>
</tr>
<tr>
<td>$G^k_g$</td>
<td>The $k$-connectivity spammer group graph of group $g$</td>
</tr>
<tr>
<td>$R_p$</td>
<td>Reviewer set of product $p$</td>
</tr>
<tr>
<td>$P_p$</td>
<td>Product set reviewed by $r$</td>
</tr>
</tbody>
</table>
4. SPAMMER GROUP DETECTION

Although the loose spammer group better reflects the mechanism of real-life group spammers, it is very difficult to spot such loosely connected review spammers. In this section, we demonstrate how to obtain such loose spammer groups via bipartite graph projection technique.

4.1. Bipartite graph model of review data

In bipartite graphs, the nodes of the graph are split into two groups, say X and Y, and the edges must connect a member of one group to a member of the other group. Bipartite graphs are widely used to capture the relationships between two kinds of objects, such as users and their interests, terms and queries, as well as many others [14, 16, 17].

In the context of review system, the two groups of a bipartite graph correspond to reviewer set and product set, and the edges correspond to the review set. If a reviewer writes a review for a product, then there is an edge connecting that reviewer and that product. Note that a reviewer may write many reviews for a particular product, we can pick a typical one or the average of the reviews as the edge, according to the situations in question. Figure 1a shows a bipartite graph consisting of four reviewers and three products. We can see that reviewers R2 and R3 review all the target products, while reviewers R1 and R4 only review products P1 and P3.

To capture the relationships among the objects in one group of the bipartite graph, the one-mode projection is often taken over the bipartite graph. A one-mode projection, also called a bipartition, involves constructing a new graph $G = (V, E)$, where $V = X$ (for X-projection), and for any $x_1, x_2 \in X$, edge $e = (x_1, x_2) \in E$ if and only if $x_1$ and $x_2$ have at least one common neighboring Y node in the original bipartite graph. In the weighted bipartite projection graph, a weight is attached to each edge of the graph to indicate the connection intensity between two objects. The weight of an edge is often naturally calculated by the number of common Y nodes of the two X nodes.

In review spammer group detection problem, we concentrate on the reviewer node projection. Figure 1b illustrates the reviewer-projection of the review bipartite graph in Fig. 1a.

Theoretically, the whole reviewer projection graph can be treated as a loose spammer group with an extremely low SS value. In fact, due to the small world phenomenon, almost all the reviewers are connected in the graph. To discover real spammer groups in the reviewer bipartition, we have to decompose the graph into meaningful small pieces, each having a high group spam score. To evaluate the spamicity of a spammer group, we develop several group spam indicators to measure the degrees of spamming activities.

4.2. Group spam indicators

Previous researchers have proposed numerous spam indicators to evaluate the spamicity of individual or group spammers for modeling or learning. These indicators include linguistic indicators, behavioral indicators, collusive indicators etc. However, these indicators are not applicable to our problem due to the different context of data. For example, group spamming indicators proposed in [11, 12] are based on FIM-generated candidate groups, which do not consider time constraints, and a reviewer might belong to many candidate groups, resulting in a large number of groups. While our proposed loose spammer groups are heavily constrained by time, a spam reviewer must belong to one and only one loose spammer group. Therefore, we designed our own spam indicators emphasizing the structural and behavioral features of the loose spammer group. All indicators are normalized to [0, 1], a larger value indicates a more spamming activity. Xu et al. [12] summarized many state-of-the-art spam indicators and analyzed the effectiveness of these indicators on a popular Chinese E-commerce website. According to their study, colluders generally express in a similar way with ordinary reviewers, or they behave normally in the linguistic level, so that linguistic indicators often underperform in discriminating spammer/non-spammers. Therefore, we do not consider linguistic indicators in this study.

1. Review tightness (RT): given a spammer group $g$, the tightness of a group $T(g)$ defined in Section 3 can naturally be a spam indicator. However, this indicator is susceptible to the number of spammers and the number of products in a group. In other words, groups with fewer members and target products are more likely to be formed by coincidence rather than on purpose. For example, if two reviewers reviewed a popular product, it is more likely that it is by accident rather than that they collude with each other to write fake reviews. On the contrary, if a large number of people write many reviews toward several products, it strongly suggests that this is a group spamming activity. The logistic function is often used to model this phenomenon, as shown in Fig. 2. To reduce the contingency of small-sized groups, we define the degree of uncontingency of group $g$ as:

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

FIGURE 1. The bipartite view of review data.
Detecting Review Spammer Groups via Bipartite Graph Projection

For the minimal spammer group with only two reviewers and one product, $L(g)$ is 0.5, for large groups, $L(g)$ asymptotically approaches 1.0. We then define the review tightness indicator of $g$ as

$$RT(g) = \frac{|V_g|}{|R_g||P_g|}L(g)$$  \hspace{1cm} (3)

2. Neighbor tightness (NT): for loose spammer groups, their $RT$ values can be very low. The neighbor tightness indicator of group $g$ considers the pairwise interreviewer tightness. We use the Jaccard similarity of product sets reviewed by two reviewers to measure the collusive behavior of two reviewers, and then take the average value over all reviewer pairs.

$$NT(g) = \text{avg}_{r_1,r_2 \in R_g} \frac{|P_{r_1} \cap P_{r_2}|}{|P_{r_1} \cup P_{r_2}|}$$  \hspace{1cm} (4)

3. Product tightness (PT): for tight spammer groups, many group members concentrate on a certain number of products, and if these reviewers do not review any other products, they are most likely to be a spammer group. Given a spammer group $g$, the product tightness of $g$, denoted by $PT(g)$, is defined as the ratio of the number of common products reviewed by all the members in $g$ to the total number of products reviewed by all the members in $g$:

$$PT(g) = \frac{|\cap_{r \in R_g} P_r|}{|\cup_{r \in R_g} P_r|}$$  \hspace{1cm} (5)

4. Average time window (TW): group spammers are likely to post fake reviews during a short-time interval, while randomly formed reviewer groups intend to span a longer time period. Given a spammer group $g$, and a product $p \in P_g$, we define the time window spamicity of $p$ as

$$TW_p(g,p) = \begin{cases} 1 - \frac{L_p - F_p}{T} & , \quad L_p - F_p \leq T \\ 0, & L_p - F_p > T \end{cases}$$

where $L_p$ and $F_p$ are the last review date and first review date of product $p$ reviewed by reviewers in $R_g$, $T$ is a user specified time threshold, say, 30 days. Then we define the TW indicator of $g$ as the average time window spamicity of all products in $g$:

$$TW(g) = \text{avg}_{p \in P_g}TW_p(g,p)L(g)$$  \hspace{1cm} (6)

Again, we use logistic curve to lower the impact of small-sized groups.

5. Rating variance (RV): group members aim to promote or demote target products, so their rating scores intend to be similar or identical. Let $S^2(p,g)$ be the variance of the rating scores of product $p$ by reviewers in $g$, we take the average variance of all target products, and use logistic curve to restrict it to the range $[0.5, 1)$. The larger the variance, the lower the spamicity. Therefore, we should transform the variance degree $[0.5, 1)$ into spamicity degree $[0, 1]$. We also use logistic curve to reduce the impact of small-sized groups:

$$RV(g) = 2 \left( \frac{1}{1 + e^{-\text{avg}_{p \in P_g}S^2(p,g)}} \right) L(g)$$  \hspace{1cm} (7)

6. Product reviewer ratio (RR): we observed that if a product is mainly reviewed by the reviewers in $g$, then the group is more likely to be a spammer group, while in randomly formed reviewer groups, the reviewed product has many more reviewers other than reviewers in $g$. We define this indicator as the average value of the ratio of the number of reviewers in $R_g$ who review product $p$ to the number of all the reviewers of $p$, $p \in P_g$:

$$RR(g) = \text{avg}_{p \in P_g} \frac{|R_{gp}|}{|R_p|}$$  \hspace{1cm} (8)

7. Early review (ER): review spammers usually review early to impact the upcoming reviewers [7]. For each product in $P_g$, we calculate its early review degree. Then the early review degree of group $g$ is defined as the average early review degree of all target products:

$$ER(g) = \text{avg}_{p \in P_g} ER_p(g,p)L(g)$$

$$ER_p(g,p) = \begin{cases} 1 - \frac{L_p - A_p}{T'}, & L_p - A_p \leq T' \\ 0, & L_p - A_p > T' \end{cases}$$  \hspace{1cm} (9)

where $A_p$ is the date when the first review of product $p$ is launched, $L_p$ is the latest date of review for product $p$ posted by any reviewer in group $g$. $T'$ is a time threshold which is set to 180 days in our experiments. ER also uses logistic curve to eliminate the impact of small groups.
8. **Group size (GS):** large groups involve many reviewers reviewing many products simultaneously, which is often premeditated. Small-sized groups, however, often come into being by accident thus are of little importance. We use logistic function to define the group size indicator.

\[
GS(g) = \frac{1}{1 + e^{-(|R_g| - 3)}},
\]

(10)

Since the minimum number of reviewers in a group is 2, the range of GS is about (0.27, 1).

Note that RT, NT, PT, RR and GS are all structural indicators and TW, RV, ER are behavioral indicators, both reveal the transaction data thus cannot be easily fabricated. We do not use linguistic indicators such as content similarity, word frequencies etc., because such review content based indicators are less discriminative due to the fact that review spammers can imitate such behaviors like genuine reviewers. By no means do we claim that if one of these indicators is large enough, then we can conclude that it is a spammer group. We need to investigate each of these indicators and take a comprehensive view of all the indicators to determine the spamicity of a group, which is exactly what the group spam scoring function SS(g) = f (g) is doing.

### 4.3. Spammer group detection algorithms

By definition, collusive behaviors among review spammers are implied by the common target products they review. To obtain the relationship between each pair of reviewers, the bipartite view of review data should be mapped into the reviewer projection graph (or the reviewer bipartition). How to efficiently construct the weighted reviewer bipartition and find all the k-connectivity spammer groups is non-trivial. In [18], an efficient algorithm was proposed to produce a k-neighboring connectivity graph which is a weighted bipartite projection graph with all weights greater than or equal to k. However, our problem is to find all the k-connectivity spammer group graph from multiple levels of k, i.e. to mine all the k-connectivity spammer group graphs for all k values, \( k \geq 1 \). So the algorithm in [18] is not applicable to our problem.

To efficiently mine all the k-connectivity spammer groups, we design a two-phase spammer group detection algorithm. In the first phase, we construct the reviewer bipartition graph from the original bipartite graph of review data; in the second phase, we search for every connected component in the reviewer bipartition as a 1-connectivity spammer group, and mine all the k-connectivity spammer groups recursively by increasing k. If a spammer group satisfies predefined maximum group size and its group spam score exceeds the predefined minimum group spam score, the group is output.

**Algorithm 1 Group Spam detection via Bipartite Projection (GSBP)**

**Input:**
- \( B \): a bipartite graph representing reviewers, products and reviews;
- \( \tau \): review time window

**Output:**
- Spammer groups ordered by spamicity score

**Description:**

1. Initialize the reviewer bipartition graph using reviewer set of \( B \), set all weights to 0;
2. for each product \( p \) in \( B \)'s product node set do
   3. for each \( p \)'s reviewer pair \((r_1, r_2)\) do
      4. if \( r_1 \) and \( r_2 \) review \( p \) in less than time interval \( \tau \) then
         5. Increment the weight of edge \((r_1, r_2)\) by 1;
   6. end if
5. end for
8. for each \( g^1 \)-connected component as \( g \) in reviewer bipartition do
   9. FindGroups(\( g^1 \), 1); // finds all k-connectivity spammer groups from \( k = 1 \)
10. end for
11. Output all spammer groups in descending order of group spam score;

Algorithms 1 (GSBP) and 2 are self-explanatory. GSBP first constructs the 1-connectivity spammer group graph \( g^1 \), then invokes Algorithm 2 to mine all the k-connectivity spammer groups recursively, and rank all the spammer groups in descending order of group spam scores. Line 1 initializes the reviewer projection graph, setting all edge weights to 0. Lines 2–8 iterate through the product set, calculating the weights of edges in the reviewer bipartition by adding 1 to the weight between two reviewers \( r_1 \) and \( r_2 \) as long as they have reviewed a common product \( p \) within the time window \( \tau \). After this, the reviewer projection graph is constructed. Then in lines 9–11, for each \( g^1 \)-connected component in the reviewer bipartition, Algorithm 2 is invoked to mine all the spammer groups which have no more than MAXSIZE reviewers and whose group spam scores exceed the minimum threshold \( \delta \). Finally, all spammer groups are output in descending order of group spam score.

Algorithm 2 exploits a divide and conquer strategy to gradually break down the whole giant reviewer group into small spammer groups. If the size of a spam group \( g.size \) is smaller than or equal to predefined maximum group size \( MAXSIZE \), and the group spam score \( g.spam \) is greater than or equal to the predefined threshold \( \delta \), then the group is output. Otherwise, the group will be split into \( g^{k+1} \)-connected components and further processed recursively.
Algorithm 2 FindGroups(g, k)
Input:
g: spammer group;
k: k-connectivity threshold;
Output:
candidate spammer groups
Description:
1: if g.size > MAXSIZE then
2: for each g^{k+1}-connected component g’ in g do
3: if g’.size > MAXSIZE then
4: FindGroups(g’, k + 1);
5: else if g’.spam ≥ δ then
6: Output g’;
7: else if g’.size > 2 then
8: FindGroups(g’, k + 1);
9: end if
10: end for
11: else if g.spam ≥ δ then
12: Output g
13: else
14: for each g^{k+1}-connected component g’ in g do
15: if g’.spam ≥ δ then
16: Output g’;
17: else if g’.size > 2 then
18: FindGroups(g’, k + 1);
19: end if
20: end for
21: end if

In Algorithm 2, whenever a connected component of G^k under level k + 1 is produced, a key problem is to determine g^{k+1}s R_g, V_g and P_g, so that g^{k+1}s spam indicator values can be calculated. R_g is just the node set in G^{k+1}, P_g is the set of all the products that being reviewed by at least two reviewers in R_g within time window τ, and V_g is the set of all the reviews that are used to construct the weighted reviewer bipartition projection graph.

Here, we give a theoretical analysis of the time complexity. The first phase is to construct the bipartite projection graph, which involves searching through every combination of two reviewers who review a common product. Let n be the number of products, deg_p be the degree of product p (number of reviewers), d_i be the i-th value of all the distinct D product degrees and n_i be the number of products having degree d_i, then the time complexity is

\[ T(n) = \sum_{p=1}^n \left( \frac{\text{deg}_p}{2} \right) = \sum_{i=1}^D n_i \left( \frac{d_i}{2} \right) \]

Fortunately, degrees of product nodes in review data usually follow the power law distribution [1], i.e. for small d_i, which dominate the degree values, their n_i-s are relatively large, and for large d_i, their n_i-s are extremely small. Hence T(n) is approximately linear with n. Figure 3 shows the log–log plot of the product degree distribution in our reference dataset, where D = 288, 2 ≤ d_i ≤ 1235. Since T(n) is determined by different n_i-s and d_i-s, we only give the simplified worst time complexity of T(n), when all d_i-s follow a uniform distribution, i.e. n_1 = n_2 = · · · = n_D = n/D,

\[ T_{\text{max}}(n) = \frac{n}{D} \sum_{i=1}^D \left( \frac{d_i}{2} \right) \approx O(nd^2) \]

where d^2 is the average value of all d_i^2-s.

The time complexity of the second phase, mostly dominated by Algorithm 2, which is an efficient recursive algorithm, is O(m log m), where m is the number of reviewers in reviewer bipartition.

4.4. The impact of time window

The time window τ is a user specified parameter which plays an important role in constructing the reviewer bipartition, which indirectly affects the final detected spammer groups. Given two different time windows t_1 and t_2, t_1 < t_2, then there exists seven relationships between two spammer groups generated under t_1 and t_2, as shown in Fig. 4. They are

1. \text{Not change}: a spammer group identified under t_1 is still a spammer group identified under t_2;
2. \text{Grow}: new reviewers are included in the spammer group under t_1 and form a new spammer group under t_2;
3. \text{Merge}: two spammer groups under t_1 merge to form a new spammer group in t_2;
4. \text{Grow and merge}: two spammer groups under t_1 merge and at least one group has taken in new reviewers to form a new spammer group under t_2;
5. \text{Break}: a k-connectivity spammer group under t_1 breaks as k increments, forming new (k+1)-connectivity spammer groups under t_2;
5. EXPERIMENTAL STUDY

5.1. The reference dataset

We use Amazon review dataset\(^2\) crawled in 2006 as a reference dataset, which was also used in \([1, 7, 10, 11]\) for review spam detection. The dataset contains information about reviewers, review text, ratings, products etc. As the whole dataset is extremely large, we only extracted the book review data from the dataset. We removed those reviewers who have less than 3 reviews since such reviewers are immense in size and often have little clues of spamming activity. We removed 6 reviewers who have reviewed more than 2000 products as such reviewers are not likely to be normal reviewers. Products with only one reviewer were also removed since such products are not subject to group spamming. The final book dataset contains 147,376 reviewers, 1,235,772 reviews and 192,357 products.

5.2. Human evaluation

Review spammer detection problem is considered to be very challenging due to the fact that there is no ground-truth spam/non-spam dataset available for model building or evaluation of proposed methods. Previous studies rely on manually labeling of gold-standard datasets. Mukherjee et al. \([11]\) and Xu et al. \([12]\) use FIM to generate candidate spammer groups to be evaluated by expert human judges. They assume each group must have at least two reviewers and work on at least three products. In \([11]\), 7052 candidate spammer groups were generated and only 2431 of them were labeled by 8 human judges. In \([12]\), 8915 groups were found for the same FIM setting. Nonetheless, reviewers in candidate spammer groups rather than the groups themselves are labeled as spammers or non-spammers.

Our proposed method is unsupervised and does not require any labeling of data. However, to evaluate the effectiveness of our ranking algorithm, human evaluation of ranking result is also imperative. Lacking essential information to determine the importance of each indicator, the safest way is to use the average value of the eight indicators to represent the group spamming score. Only 1189 candidate spammer groups were found by GSBP using \(\tau = 30\) days, \(\delta = 0.4\), MAXSIZE = 50. Although the minimal spammer group in our case consists of at least two reviewers and one product, the number of candidate groups is significantly reduced in comparison to FIM-generated groups, because the spammer group by our definition is heavily time constrained. Of all the 1189 groups, only 5 groups have tightness value \(T(g) = 1\), which can also be detected by FIM-based methods.

As reported by \([11]\), labeling fake reviewer groups is much easier than labeling individual review spammers, due to the fact that spammer groups give good context for judging and comparison. We employed three post-graduate students who are very familiar with E-commerce environment to fulfill the evaluation task. Benefiting from previous spam detection guidance along with our own observations, we tried our best to minimize the human bias in evaluation process. Like \([7]\), we also developed a data visualization software to reveal the relationship among reviewers, products and statistics of candidate spammer groups, which can greatly help human evaluators to determine the spamicity of a group. The software can also provide a hyperlink to the reviewer’s online profile at Amazon.com so that more clues other than the information in the reference dataset could be found to determine the spamicity of a group. Finally, we got a gold-standard labeled dataset with 388 spammer groups and 801 non-spammer groups.

5.3. Empirical analysis of group spam indicators

With our labeled dataset of review spammer groups, we can perform an empirical analysis of our provided eight group spam indicators. Cumulative distribution functions (CDF) were used to show the discrimination capability of spam indicators \([11–13]\). Figure 5 plots the CDFs of the eight group spam indicators, i.e. the cumulative percent of spammer groups (solid) and non-spammer groups (dashed) vs. the group spam indicator values. The larger the gap between the curves of the two distributions, the better discrimination capability the

---

\(^2\) http://liu.cs.uic.edu/download/data/
indicator will achieve. We can see that all the eight indicators can to some extent discriminate spam/nospam behaviors. In general, structural indicators such as neighbor tightness (NT), product tightness (PT) and review tightness (RT) perform the best, behavioral indicators such as rating variance (RV), early review (ER) and average time window (TW) also perform well. Note that group size (GS) performs the least, which indicates that group size is not very much correlated with the spamicity of a group.

5.4. Performance on reference dataset

Since there is no previous work aiming to find loose spammer groups, and no state-of-the-art machine learning algorithm is
applicable to this problem, we evaluate the effectiveness of our proposed methods by human evaluation. Human evaluation is widely adopted in IR and opinion spamming research area due to the lack of ground-truth data [1, 2, 7, 9, 11, 12]. Because our method produces the ranking of review spammer groups, one can conjecture that groups that lie on the top positions will be more suspicious to be spammer groups. Figure 6 shows the detecting precision, recall and F1 values at the top n positions. We can see that the precision is very high when n is small, and degrades as n gets larger. The recall asymptotically approaches 1 as n increases. F1 value suggests that the best tradeoff between precision and recall is when n is between 300 and 400.

Figure 7 shows the number of k-connectivity spam groups for different k values. From the figure we can see that most of the spam groups are generated when k is small. As we gradually increase k, many more spam groups are identified.

To further study the efficiency of our proposed algorithm, we chronologically divide our dataset into six equal-sized sub-datasets based on the number of products (reviewers), and then construct six new datasets with the number of products (reviewers) equals to n times of the size of the sub-datasets, n = 1, 2, . . . , 6. Figure 8 illustrates the execution time of our proposed algorithm as the number of products or reviewers increases. From the figure we can see that the time is almost proportional to the number of products or reviewers, which verifies our theoretical analysis in Section 4.3.

5.5. Impact of parameters

There are three parameters in our proposed algorithms: τ, δ and MAXSIZE. To evaluate the impact of these parameters, we use different combinations of the three parameters to produce ranking results, and investigate the impact of these parameters.

First, we fix τ = 30 days, and set MAXSIZE = 50, 40, 30, 20 reviewers, δ = 0.4, 0.5, 0.6. Table 2 shows the number of
TABLE 2. Result set comparison: # of groups/# of reviewers w.r.t MAXISIZE and δ (τ = 30).

<table>
<thead>
<tr>
<th>δ</th>
<th>MAXISIZE 20</th>
<th>MAXISIZE 30</th>
<th>MAXISIZE 40</th>
<th>MAXISIZE 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>1196/3440</td>
<td>1192/3510</td>
<td>1189/3515</td>
<td>1189/3517</td>
</tr>
<tr>
<td>0.5</td>
<td>505/1534</td>
<td>506/1559</td>
<td>503/1564</td>
<td>503/1566</td>
</tr>
<tr>
<td>0.6</td>
<td>181/568</td>
<td>182/590</td>
<td>182/590</td>
<td>182/590</td>
</tr>
</tbody>
</table>

TABLE 3. Number of groups (reviewers) for different time windows (δ = 0.4, MAXISIZE = 50).

<table>
<thead>
<tr>
<th>Type</th>
<th>τ = 30</th>
<th>τ = 45</th>
<th>τ = 60</th>
</tr>
</thead>
<tbody>
<tr>
<td># of groups</td>
<td>1189</td>
<td>1299</td>
<td>1340</td>
</tr>
<tr>
<td># of reviewers</td>
<td>3517</td>
<td>3838</td>
<td>3954</td>
</tr>
</tbody>
</table>

spammer groups and the corresponding number of reviewers of all the groups for each combinations of MAXISIZE and δ. From the table, we can see that the number of returned groups is not sensitive to MAXISIZE. This is because the number of reviewers in a group is inherently determined by the relations of reviewers, products and review dates. δ can severely impact the number of groups (reviewers) because a higher δ results in highly suspicious spammer groups, leading to more accurate results. Note that the number of reviewers from all groups varies much more than the number of total groups, because reviewers in groups might also change, but is still relatively stable.

To study the impact of τ, we fix MAXISIZE = 50, δ = 0.4 and set τ = 30, 45, 60 days. Table 3 shows the number of spammer groups and the corresponding number of reviewers included in all the groups. We can see that a larger τ leads to more spammer groups (reviewers). We also counted the number of groups of each kind of relations by two different τs, as shown in Table 4. We can see that most of the groups remain not change, then follow new and grow, other relations are rare. This reveals that the impact of time window τ is relatively small. Moreover, those groups which remain unchanged when time window changes from τ₁ to τ₂ are not sensitive to time window, thus should be given higher priority to be taken as spammer groups.

5.6. Comparison with FIM candidate groups

As our bipartite projection method aims to replace FIM to generate candidate spammer groups, we also compared the candidate groups discovered by the two methods. We used FPgrowth³ as a FIM tool to mine all the maximal frequent itemsets. Although we used minimum support = 3 and minimum itemset length = 3, representing each candidate group has at least 3 reviewers and all reviewers at least commonly review 3 products, we still got 107 305 candidate groups, which is far more than the number of candidate groups generated in [11, 12]. Since we use different datasets, this indicates our book review dataset involves more co-reviewing.

To compare, we also calculated the eight indicator values for each of these groups using the same setting as used in GSBP to generate candidate groups for human evaluation (τ = 30). We finally got 44 112 groups (61 026 reviewers) with average indicator value ≥0.4, far more than the number of groups detected by GSBP, which is 1 189 groups (3 517 reviewers). Figure 9a and b shows the box-and-whisker plots of the indicator value distributions of the groups detected by the two methods. We can see that, although the average indicator values of the two methods are approximately the same, their indicator value distributions are quite different. Specifically, the NT, PT values by FIM are extremely low, because many reviews in the group are excluded due to time window limit. The TW value is also very low and RR values are too high, while all indicator values by GSBP are well distributed. The distributions of GS indicator demonstrate that FIM tends to return large groups. This is because FIM does not take into account time constraint in generating candidate spammer groups, so a large number of irrelevant reviewers are included in the result set. Moreover, groups detected by FIM are overlapping, i.e. the same reviewer(s) can be members of multiple groups, resulting in large number of candidate groups. In comparison, groups detected by GSBP have no intersection.

We also fetched the top 1 515 groups (3 519 reviewers) from FIM candidates which comprise comparative reviewers to the number of reviewers detected by GSBP and plot its distribution in Fig. 9c. We can see that the distribution of indicators gets even worse. In general, GSBP can generate more meaningful spammer groups than FIM. Therefore, our method can also be used as an alternative data preprocessing technique in [11–13] by tuning parameters to generate desired number of candidate groups.

³ http://www.borgelt.net/fpgrowth.html

TABLE 4. Number of groups for different kind of relations in groups identified under time window τ₂ compared with time window τ₁ (δ = 0.4, MAXISIZE = 50).

<table>
<thead>
<tr>
<th>Relation</th>
<th>Time window</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30–45</td>
</tr>
<tr>
<td>Not change</td>
<td>812</td>
</tr>
<tr>
<td>New</td>
<td>312</td>
</tr>
<tr>
<td>Grow</td>
<td>110</td>
</tr>
<tr>
<td>Break</td>
<td>35</td>
</tr>
<tr>
<td>Merge</td>
<td>12</td>
</tr>
<tr>
<td>Break and grow</td>
<td>11</td>
</tr>
<tr>
<td>Grow and merge</td>
<td>7</td>
</tr>
</tbody>
</table>

Downloaded from https://academic.oup.com/comjnl/article-abstract/59/6/861/2568618 by guest on 10 December 2018
5.7. Case study

To demonstrate the quality of our detected loose spammer groups, here we give five typical groups identified by our algorithm, as well as the online profile addresses of all the reviewers. By no means do we claim that these groups are definitely real spammer groups. Instead, we state that these groups are highly suspicious and should be further analyzed by review website administrators. Note that not all reviews in our dataset still exist on Amazon.com due to the fact that Amazon might update or remove their review data from time to time.

(i) Group 1: this group contains three reviewers. Each reviewer reviewed three common books (cooking series) by the same author within 2 days, all rating 5 stars. The helpfulness of reviewers on Amazon are 100, 100 and 96%, respectively. According to [1], review helpfulness can also be spammed. Such high helpfulness often implies spamming activity. Among the three reviewers, only one reviewer posted another two reviews in 2012 and 2015, the other two did not post any reviews any more. This strongly suggests that this is a spam group aiming to promote the author’s books. Although this group can also be detected by FIM, irrelevant reviewers might be included, e.g. a reviewer who also reviewed these three books 3 months later.

(ii) Group 2: this group contains six reviewers. All the six reviewers reviewed three common books within 16 days, and most of the reviewers reviewed the three books in 1 day except one reviewer, who reviewed in 2 days. All ratings are five stars, and the helpfulness of reviewers are all very high. Three reviewers involve duplicate or near duplicate review texts, and one reviewer whose member id is A1OENO52VAF8I8 involves inconsistent review text with the corresponding book. All reviewers did not review any other books after that.

(iii) Group 3: this group contains only two reviewers and one book, which might not be able to be identified by FIM approach, due to the combinatorial explosion when min support is <3. Note that the two reviewers come from the same city. The first reviewer reviewed the same book four times in 6 days, and the review texts are very similar. The second reviewer reviewed the same book three times in 2 days, and two out of three review texts are identical. All ratings are five stars.

(iv) Group 4: this group contains five reviewers who commonly reviewed two books by the same author, which is also impossible to be identified by FIM technique. Almost every reviewer in the group involves multiple reviewing a book. The time span is very narrow, and the rating scores are all five stars. All reviewers became inactive after that.

http://www.amazon.com/gp/pdp/profile/AA7ZUMTN8SWCC,A3VW4Y8S2BQ67Y,A20K59YEF3LT
5 http://www.amazon.com/gp/pdp/profile/A1QB56CL3PQVPA1OENO52VAF8I8,A1ZT925Y44SE5,AQLM33J521CN,A1A777UX97MRQ6
6 http://www.amazon.com/gp/pdp/profile/ATS7Z50QX9IOS,A1E51XHXNEVCEH
7 http://www.amazon.com/gp/pdp/profile/A3NPIFL85G0LNU,A1DG02WWB5YFW,ADAP8UD2XN5S,AQGBTJ1YAPH9D,A3UUPKJWFICM2Q
(v) Group 5: this group contains 10 reviewers and 6 target products. Each member of the group reviewed 3–5 books (some books are of the same author) in less than a week, rating 4–5 stars, and each book was reviewed by 2–10 reviewers of the group. Some review texts among different reviewers are quite similar to each other. The 10 reviewers did not review any other book. All this evidence strongly suggests that this is a well-organized spamming group aiming at promoting their products. This is a typical loose spammer group which is very difficult to be detected by FIM-based methods.

6. CONCLUSION AND FUTURE WORK

Online review systems are subject to opinion spamming. Detecting fake reviews or reviewers is becoming an urgent issue. In this paper, we propose the loose spammer group detection technique using bipartite graph projection. Unlike FIM-based method, our approach can find loosely connected spammer groups which are frequently occurring in realworld review websites. By exploiting effective group spam indicators to evaluate the spamicity of detected groups, a divide and conquer algorithm is designed to efficiently detect and rank loose spammer groups with high precision and recall. As future work, we plan to investigate other group spamming evaluation methods to improve the precision and recall, and seek new methods to overcome the impact of predefined parameters such as time windows, minimum group spam score and maximum group size. Besides, we also plan to incorporate our method with existing FIM-based group spamming detection techniques.

ACKNOWLEDGEMENTS

We thank Prof Daling Wang, Dr Xiaodong Li and Rui Feng for their valuable suggestion or the anonymous reviewers for insightful comments.

FUNDING

This research was partially supported by National Science Foundation of China under Grant No. 61373159.

REFERENCES


