Combating Product Review Spam Campaigns via Multiple Heterogeneous Pairwise Features

Chang Xu* Jie Zhang*

Abstract

Spam campaigns spotted in popular product review websites (e.g., amazon.com) have attracted mounting attention from both industry and academia, where a group of online posters are hired to collaboratively craft deceptive reviews for some target products. The goal is to manipulate perceived reputations of the targets for their best interests. Many efforts have been made to detect such colluders by extracting pointwise features from individual reviewers/reviewer-groups, however, *pairwise* features which can potentially capture the underlying correlations among colluders are either ignored or just explored insufficiently in the literature. We observed that pairwise features can be more robust to model the relationships among colluders since they, as the ingredients of spam campaigns, are correlated in nature. In this paper, we explore multiple heterogeneous pairwise features in virtue of some collusion signals found in reviewers' rating behaviors and linguistic patterns. In addition, an unsupervised and intuitive colluder detecting framework has been proposed which can benefit from these pairwise features. Extensive experiments on real dataset show the effectiveness of our method and satisfactory superiority over several competitors.

1 Introduction

Online product reviews nowadays have become increasingly valuable for consumers to make their purchase decisions, which has largely motivated a group of paid professionals to fabricate "sound genuine" reviews for product reputation manipulation. It is reported that the volume of possibly fictitious reviews on yelp.com rose from 5% in 2006 to 20% in 2013 [9]; remarkably about 1/3 of consumer reviews are estimated to be suspicious, if not forged, on the Internet [15]. A growing number of online businesses have been driven by the desire to achieve positive public impression through receiving online praises from their consumers, causing the formation of a huge shady market for crafting phony online product reviews.

Review spammers today are increasing in sophistication for evading detection, e.g., by forming *collaborative coalitions*. By penetrating into a real spam campaign, Chen et al. [2] found some online paid posters forming professional organizations where participants have their own roles, e.g., project managers, trainers, and posters. Mukherjee et al. [11] spotted spammer groups in amazon.com who collaboratively write fake reviews to promote/demote some target products. Xu et al. [18] revealed analogous behaviors in Chinese review websites; they found that such group spammers tend to establish tiny collusive groups to further evade detection. It is worth noting that spam coalitions should be more harmful than individual spammers; not only can they easily dominate the sentiments towards target products via flooding deceptive opinions, but they can also "hide" their suspicious behaviors by balancing workload within spam campaigns.

This paper also focuses on detecting group colluders in online product review spam campaigns. However, unlike previous studies [11, 18] which hinge on pointwise features applicable to individual reviewers/reviewer-groups only, we adopt a different path where pairwise relationships among colluders are best explored and leveraged for detection. For example, the comparison of two colluders from the same spam campaign may reveal their affinitive behaviors such as explicitly reviewing the same items, expressing similar opinions on these items, or implicitly leaving proximate temporal traces incurred by their following up the campaign schedule. Compared to group-based pointwise features, pairwise features are more fine-grained to directly reflect the relations among colluders. Hence instead of detecting individual spammers/spam groups, we propose to detect spam pairs. The proposed heterogeneous pairwise features can also complement each other in practice, producing a more robust model for correlating colluders.

In a nutshell, this paper offers the following contributions: 1) to the best of our knowledge, pairwise features are first explicitly utilized to detect group colluders in online product review spam campaigns, which can reveal collusions in spam campaigns from a more fine-grained perspective; 2) a novel detecting framework named FraudInformer is proposed to cooperate with the pairwise features which is (i) intuitive - the detection task is formulated as an autonomous process of mutual disclosure at a virtual trial where colluders are supposed to be capable of report each other by themselves, (ii) unsupervised - no prior knowledge of spam annotations is needed; 3) our method has been evaluated on real dataset and extensive experimental results

*School of Computer Engineering, Nanyang Technological University. (Email: xuch0007@e.ntu.edu.sg, zhangj@ntu.edu.sg)

show that it outperforms both baselines and existing unsupervised colluder detection approaches.

2 The Proposed Framework - FraudInformer

This section elaborates the proposed unsupervised framework for colluder detection. Given a typical review website (e.g., amazon.com) containing a set of products $\mathcal{P} = \{p_k\}$ with each commented by a list ℓ_k of reviewers in chronological order, the goal is to rank all the reviewers $\mathcal{V}$ in the website globally so that top-ranked ones are more likely to be colluders. For each reviewer $v \in \mathcal{V}$ we associate a global spam score or *spamicity* s which specifies the suspectedness of his/her engaging in the collusion of spam campaigns. The spamicities of colluders are supposed to be higher than those of non-colluders which are used for producing the ranking.

2.1 Mutual Disclosure Modeling The basis of FraudInformer relies on the idea of *mutual disclosure* inspired by the concept of criminal trials where a suspect is asked to give his/her accomplices away. Similarly in our context, colluders participating in a spam campaign are also accomplices of each other. Given the behavioral clues collected from the reviews created by colluders as the evidence against their collusions, it is possible to expose them all at once by proceeding with proper "criminal trials" against them.

Specifically, we propose the "criminal trial" scheme that for each product p_k reviewed by a chronological list ℓ_k of n_k reviewers, each reviewer v_i located at position $\ell_k(i)$ on ℓ_k is asked to accuse other surrounding reviewers on ℓ_k within an *investigating range* ζ, based on the temporal locality property of spammers [3]. The set of reviewers to be accused by v_i on ℓ_k is $R_{k,i}(\zeta) = \{v_j || \ell_k(i) - \ell_k(j)| \leq \zeta, j \neq i\}$. For generality purpose, ζ can be any functional realization of the temporal locality, e.g., ζ can be a function of n_k: $\zeta(n_k) = \sqrt{n_k}$. One merit of this scheme is that it avoids the computation of all possible pairs of reviewers in a website in which case most of the pairs can hardly be colluders if they have not co-reviewed any common product; the computational complexity has thus been reduced from $O(|\mathcal{V}|^2)$ to $O(2\zeta|\mathcal{V}|)$.

On the other hand, the amount of the evidence gathered during the accusation should also be considered and reflected, as the collusion evidence gathered from colluder pairs should be stronger than that from non-colluder pairs; the more intensively two reviewers collude with each other, the more evidence of their collusion can be gathered. Here we model the amount of collusion evidence (or we call collusiveness) gathered from a reviewer pair (v_i, v_j) as a real-valued symmetric function $\Phi(i, j)$. It is worth noting that the collusiveness is crucial to the whole detection framework since it serves as the only testimony during the entire judgement. Then the mutual disclosure based "criminal trial" regarding all reviewers of a site can be computationally formalized as a process of collusiveness propagation. Specifically, the spamicity s_i of reviewer $v_i \in \mathcal{V}$ is computed as the sum of the product of the amount of evidence $\Phi(i, j)$ provided by all surrounding reviewers $v_j \in R_{k,i}(\zeta)$ and their own spamicities, over all products $k \in \mathcal{P}_i$ reviewed by v_i:

$$s_i = \sum_{k \in \mathcal{P}_i} \sum_{v_j \in R_{k,i}(\zeta)} s_j \cdot \Phi(i, j) \tag{2.1}$$

During the propagation, the spamicity of a true colluder will increase for accumulating more collusiveness from his/her accomplices while a non-colluder will gain less collusiveness as (s)he is not supposed to collude with neither colluders nor other legitimate reviewers.

2.2 Confidence Weighting Despite being derived from external collusion facts (the review data), the collusion evidence provided by a particular reviewer about others may not be equally convincing internally. For example, a colluder may expect the reviews posted by his/her accomplices to be near his/her own postings due to their synchronous engagements in the same spam campaign, thus the confidence should be higher for the evidence provided about nearby reviewers. In FraudInformer, this semantic is captured by the confidence function $\Omega_j(i, R)$: to what degree the evidence provided by v_j about $v_i \in R_{\bullet,j}(\zeta)$ should be trusted from the perspective of v_j. We consider two classes of confidence functions here, i.e., symmetric and asymmetric.

Symmetric confidence functions $\Omega_j^{\dagger}(i, R)$ of v_j assume his/her equally distributed belief in the evidence about nearby reviewers $v_i \in R_{\bullet,j}(\zeta)$. The simplest realization can be based on the uniform kernel function [14] which assigns identical confidence to the evidence about each nearby reviewer:

$$\Omega_j^{\dagger U}(i, R) = \frac{1}{2}\mathbf{I}_{\{v_i \in R_{\bullet,j}(\zeta)\}} \tag{2.2}$$

where $\mathbf{I}_{\{\bullet\}}$ is the identity function. A more sophisticated case that encodes the concentration of spam reviews can be based on the Epanechnikov kernel function:

$$\Omega_j^{\dagger E}(i, R) = \frac{3}{4}\left(1 - \left(\frac{|\ell_{\bullet,i} - \ell_{\bullet,j}|}{\zeta(\bullet)}\right)^2\right)\mathbf{I}_{\{v_i \in R_{\bullet,j}(\zeta)\}} \tag{2.3}$$

which posits that evidence about closer reviewers on product reviewer lists matters more than remote ones.

Asymmetric confidence functions $\Omega_j^*(i)$ on the other hand treat the confidence of v_j in each surrounding reviewer differently. An example can be a function that only trusts the evidence about those who have the top K collusiveness $\Phi(i, j)$ with v_j:

$$\Omega_j^{*K}(i, R) = \frac{1}{K}\mathbf{I}_{\{\Phi(i,j) \geq \Phi^{(K)},\ v_i \in R_{\bullet,j}(\zeta)\}} \tag{2.4}$$

where $\Phi^{(K)}$ is the Kth largest collusiveness in the set. The intuition is that stronger evidence itself to some extent implies higher possibility of true collusion and thus should gain higher confidence.

Finally, to account for the impact of confidence weighting, the original expression (Eq. 2.1) for computing the spamicity of reviewer v_i can be modified to:

$$s_i = \sum_{k \in \mathcal{P}_i} \sum_{v_j \in R_{k,i}(\zeta)} s_j \cdot \Phi(i,j) \cdot \Omega_j(i, R_{k,j}(\zeta)) \tag{2.5}$$

2.3 Global Ranking To obtain a final ranking of reviewers, a propagation-based ranking algorithm is proposed based on the mutual disclosure model (Eq.(2.5)). Recall that at the end of mutual disclosure, a colluder will gain high spamicity for accumulating high collusiveness from the accomplices who also have high spamicities, and a non-colluder's spamicity will be low for the low collusiveness received from either colluders or other non-colluders. In other words, *reviewers who provide strong evidence of their collusion with other high-spamicity reviewers tend also to have high spamicities (i.e., giving themselves away).* Inspired by this observation, we consider to extend the Markov random walk model [13] to obtain a stabilized ranking of reviewer spamicities for its convergence property, where the spamicities of reviewers can be interpreted as the "authorities" of web pages in hyperlink structure. In the context of MRW, each directed reviewer pair $< v_i, v_j >$ is associated with a collusion weight $f(i \rightarrow j)$ which is equal to the sum of the product of the collusiveness between v_i and v_j, $\Phi(i,j)$ and the corresponding confidence of v_i in the evidence about v_j, $\Omega_i(j, R)$, over all products $\mathcal{P}_i$ reviewed by v_i:

$$f(i \rightarrow j) = \sum_{k \in \mathcal{P}_i} \Phi(i,j) \cdot \Omega_i(j, R_{k,i}(\zeta)) \tag{2.6}$$

The transition probability from v_i to v_j is defined by normalizing the corresponding collusion weight as:

$$p(i \rightarrow j) = \begin{cases} \frac{f(i \rightarrow j)}{\sum_{j'=1}^{|\mathcal{V}|} f(i \rightarrow j')} & \text{if } \sum f \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

Recall that $\mathcal{V}$ is the set of all reviewers in the site. The row-normalized transition matrix is $\mathbf{M} = [\mathbf{M}_{i,j}]_{|\mathcal{V}| \times |\mathcal{V}|}$ where $\mathbf{M}_{i,j} = p(i \rightarrow j)$. To meet the conditions of $\mathbf{M}$ being a stochastic matrix and guarantee the existence of a stationary distribution, the rows with all zeros are substituted by a smoothing vector with each element set to $1/|\mathcal{V}|$. Finally, the spamicity s_i of v_i can be formulated in a recursive manner as follows:

$$s_i = \eta \sum_{j: i \rightarrow j} s_j \cdot \mathbf{M}_{i,j} + (1 - \eta) \frac{1}{|\mathcal{V}|} \tag{2.8}$$

where $\eta \in [0, 1]$ is a damping factor to control the probability of teleportation, which is set to 0.85 in our case. The ranking of reviewers by their spamicities is obtained by running Eq.(2.8) iteratively until convergence.

3 Collusiveness Measure: Pairwise Features

In this section, we present multiple heterogeneous pairwise features to measure the collusiveness between reviewer pairs, i.e., $\Phi(i,j)$ in Eq. (2.5). Multiple observable and retrievable review data such as ratings, timestamps, text, and products/brands being reviewed are considered. It is worth noting that each dimension can only cover a portion of colluding behaviors (e.g., not all the pairs having similar ratings would necessarily imply the occurrence of collusion), a final combination is needed for robustness and comprehensiveness purpose.

Product-based Sentiment Deviation (PSD): having the same spamming goal of promoting/demoting the reputations of the targets, colluders tend to express similar opinions by giving similar ratings, which can lead to lower rating deviation among them. Given a pair of reviewers v_i and v_j, PSD computes the degree of rating deviation towards their commonly reviewed products:

$$\phi_{psd} = \frac{2}{1 + e^{d_r^{\mathcal{P}}(i,j)}} \cdot \alpha_{ij} \tag{3.9}$$

where $\alpha_{ij} = \frac{|P_i \cap P_j|}{|P_i \cup P_j|}$ is a damping factor accounting for the bias induced by the absolute difference between their respectively reviewed products; $P_{i(j)}$ is the set of products reviewed by $v_{i(j)}$. $d_r^{\mathcal{P}}(i,j)$ is the average rating deviation of (v_i, v_j) over their commonly reviewed products $P_i \cap P_j$: $d_r^{\mathcal{P}}(i,j) = \text{avg}_{p_k \in P_i \cap P_j}(|r_i^{p_k} - r_j^{p_k}|)$ where $r_{i(j)}^{p_k}$ is the rating given by $v_{i(j)}$ for the commonly reviewed product p_k.

Product-based Time Deviation (PTD): following the same predefined spam schedule, colluders prone to behave in a lockstep manner. In other words, the temporal traces of their postings tend to be close on the commonly reviewed products. Given a pair of reviewers v_i and v_j, ϕ_{ptd} evaluates the reviewing time gap on their commonly reviewed products:

$$\phi_{ptd} = \frac{1}{1 + d_t^{\mathcal{P}}(i,j)^{\lambda}} \cdot \alpha_{ij} \tag{3.10}$$

where $\lambda \geq 1$ is a parameter to accelerate the rate of decay for the average reviewing time deviation of (v_i, v_j) over $P_i \cap P_j$, $d_t^{\mathcal{P}}(i,j) = \text{avg}_{p_k \in P_i \cap P_j}(|t_i^{p_k} - t_j^{p_k}|)$ where $t_{i(j)}^{p_k}$ is the reviewing timestamp for $p_k \in P_i \cap P_j$ by $v_{i(j)}$.

Product-based Review Text Similarity (PTS): previous studies [8, 10, 11] observed that review spammers are inclined to write fake reviews with similar contents not only for saving efforts but also for expressing similar opinions. To capture this signal, given a pair of reviewers v_i and v_j, ϕ_{pts} is formulated as the maximum cosine similarity between their review texts over $P_i \cap P_j$:

$$\phi_{pts} = \max_{p_k \in P_i \cap P_j} (cosine(v_i^{p_k}, v_j^{p_k})) \cdot \alpha_{ij} \tag{3.11}$$

where $v_{i(j)}^{p_k}$ is the review text written by $v_{i(j)}$ for $p_k \in P_i \cap P_j$. Each review text is represented by a bag of bi-grams and $cosine(v, v')$ computes the cosine similarity of the bi-gram TF-IDF vectors of v and v'.

Brand-based Sentiment Deviation (BSD): brand information has been shown to be helpful [5, 7, 18]. In [18] the author found that instead of targeting at commonly reviewed products, colluders nowadays have been assigned different sets of products for reviewing.

However, all the compromised products may still have the same brands. Here we consider this "higher level" information. Given a pair of reviewers v_i and v_j, ϕ_{bsd} computes the degree of average rating deviation towards their commonly reviewed brands:

$$\phi_{bsd} = \frac{2}{1 + e^{d_{\bar{r}}^{\mathcal{B}}(i,j)}} \cdot \beta_{ij} \tag{3.12}$$

where $\beta_{ij} = \frac{|B_i \cap B_j|}{|B_i \cup B_j|}$ is a damping factor corresponding to brand; $B_{i(j)}$ is the set of brands reviewed by $v_{i(j)}$. $d_{\bar{r}}^{\mathcal{B}}(i,j)$ is the average rating deviation between v_i and v_j over $B_i \cap B_j$: $d_{\bar{r}}^{\mathcal{B}}(i,j) = \text{avg}_{b_k \in B_i \cap B_j}(|r_i^{b_k} - r_j^{b_k}|)$ where $r_{i(j)}^{b_k} = \text{avg}_{p_{k'} \in P_{i(j)}, brand(p_{k'}) = b_k}(r_{i(j)}^{p_{k'}})$ is the average rating given by $v_{i(j)}$ for $b_k \in B_i \cap B_j$.

Brand-based Time Deviation (BTD): similar to the "product" version PTD, BTD computes the degree of reviewing time gap between reviewers v_i and v_j at brand level:

$$\phi_{btd} = \frac{1}{1 + d_{\bar{t}}^{\mathcal{B}}(i,j)^{\lambda}} \cdot \beta_{ij} \tag{3.13}$$

where $d_{\bar{t}}^{\mathcal{B}}(i,j)$ measures the average deviation of the reviewing time intervals for $B_i \cap B_j$, between v_i and v_j: $d_{\bar{t}}^{\mathcal{B}}(i,j) = \text{avg}_{b_k \in B_i \cap B_j}(|\max(T_i^{b_k}) - \max(T_j^{b_k})| + |\min(T_i^{b_k}) - \min(T_j^{b_k})|)$ where $T_{i(j)}^{b_k} = \{t_{i(j)}^{p_k} | p_k \in P_{i(j)}, brand(p_k) = b_k\}$ is the set of reviewing timestamps for $b_k \in B_i \cap B_j$ by $v_{i(j)}$.

Brand-based Review Text Similarity (BTS): similar to the "product" version PTS, BTS measures the maximum review text similarity towards $B_i \cap B_j$:

$$\phi_{bts} = \max_{b_k \in B_i \cap B_j} \Big(\max_{\substack{p \in P_i, brand(p) = b_k \\ p' \in P_j, brand(p') = b_k}} (cosine(v_i^p, v_j^{p'})) \Big) \cdot \beta_{ij} \tag{3.14}$$

Reviewing Activity Homophily (RAH): for a spam campaign, the schedule or budget would affect the reviewing activity patterns of involved colluders. Specifically, these colluders tend to be busy during the campaign period while idle when no task is available. Based on this intuition, RAH captures the commonality of posting activities of a pair, v_i and v_j. We first split the global timeline into small time slots $\{s_1, s_2, ..., s_k\}$ with equal width τ. Each slot corresponds to a time period of interest. ϕ^{rah} can then be defined as:

$$\phi_{rah} = \frac{1}{1 + \left[\frac{KL(q_i \| q_j) + KL(q_j \| q_i)}{2}\right]^{\lambda}} \tag{3.15}$$

where $KL(\cdot\|\cdot)$ is the KL divergence. $q_{i(j)} = \left\{\frac{n_{i(j)}^s}{\sum_{s' \in S_i \cap S_j} n_{i(j)}^{s'}}\right\}_{s \in S_i \cap S_j}$ is the distribution of $v_{i(j)}$ over the number of reviews posted in each of their common time slots $n_{i(j)}^s$. $S_{i(j)}$ is the set of time slots within which $v_{i(j)}$ has posted at least one review.

Reviewing Lifetime Homophily (RLH): it has been reported that spammers, unlike authentic reviewers, typically do not use their accounts for too long [10]. The usage patterns of their accounts are expected to be similar with each other since they may follow a unified arrangement in a spam campaign, leading to similar time spans of their reviewing lifetimes. RLH captures the difference of the reviewing lifetimes of (v_i, v_j):

$$\phi_{rlh} = \frac{1}{1 + |LT_i - LT_j|^{\lambda}} \tag{3.16}$$

where $LT_{i(j)} = \max(T_{i(j)}^{\mathcal{P}}) - \min(T_{i(j)}^{\mathcal{P}})$ computes the lifetime of $v_{i(j)}$, and $T_{i(j)}^{\mathcal{P}} = \{t_{i(j)}^p | p \in P_{i(j)}\}$ is the set of reviewing timestamps of $v_{i(j)}$ for the products in $P_{i(j)}$.

Pairwise Feature Combination. Each pairwise feature has already been normalized within 0 and 1. We integrate all dimensions via a convex combination, to obtain the overall collusiveness measure for each reviewer pair (v_i, v_j): $\Phi(i,j) = \sum_k w_k \cdot \phi_k(i,j)$ where $\sum_k w_k = 1, w_k \geq 0$, k is the index for each pairwise feature. The weighting parameter $\mathbf{w} = \{w_k\}$ specifies the importance of each feature; bigger weights should be assigned to more effective ones. As our proposed framework is unsupervised, uninformative equal emphasis will be given to each feature dimension.

4 Empirical Analysis

Dataset. Building an annotated spam review dataset is non-trivial. We manage to obtain the dataset used in [18] where 1,937 colluders (+) and 3,118 non-colluders (–) are identified in the product reviews of Amazon.cn. The reviews for all the products reviewed by these annotated reviewers are used in our experiments, which in total involves 3,987 products, 140,258 reviewers[1], and 265,793 reviews.

Evaluation Criteria. Two well-known ranking based metrics are used for our experimental analysis [11].

1) Precision@k: defines the precision at cut-off k in the ranking list which corresponds to the ratio of colluders in the top k ranks.

2) Normalized Discounted Cumulative Gain (NDCG): evaluates resulting rankings with respect to an ideal ranking based on reviewers' spamicities, which favors rankings where colluders with highest spamicities are ranked at the top. NDCG@k is defined to be:

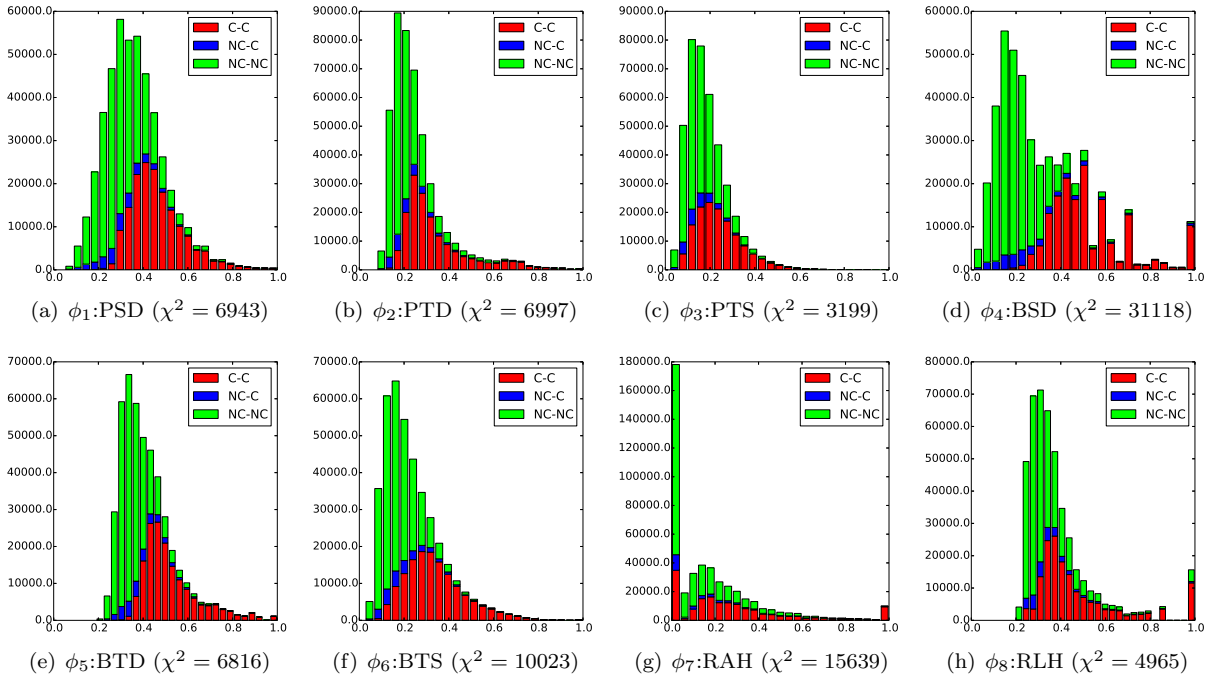
$$NDCG@k = \frac{DCG@k}{IDCG@k}; DCG@k = \sum_{i=1}^{k} \frac{2^{c_i} - 1}{log_2(1+i)} \tag{4.17}$$

where c_i is the binary class value of the reviewer (1:colluder, 0:non-colluder) ranked at position i. IDCG@k is the discounted cumulative gain (DCG) of the ideal ranking at position k where all colluders are ranked higher than all non-colluders.

4.1 Effects of Pairwise Features

As the measures of collusiveness between reviewers, the proposed multiple pairwise features are the core to differentiate colluders from non-colluders in FraudInformer. Effective pairwise features are expected to distinguish the

[1] Singletons, a special type of spammers writing only one review in their lifetimes, are not considered in our experiments since the proposed pairwise features require reviewers to have adequate reviewing histories. Studies like [17] can handle this issue.



(a) ϕ_1:PSD ($\chi^2 = 6943$) (b) ϕ_2:PTD ($\chi^2 = 6997$) (c) ϕ_3:PTS ($\chi^2 = 3199$) (d) ϕ_4:BSD ($\chi^2 = 31118$)

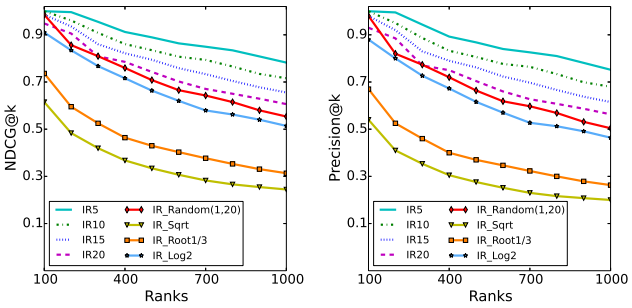
(e) ϕ_5:BTD ($\chi^2 = 6816$) (f) ϕ_6:BTS ($\chi^2 = 10023$) (g) ϕ_7:RAH ($\chi^2 = 15639$) (h) ϕ_8:RLH ($\chi^2 = 4965$)

Figure 1: Histograms of pairwise features over three types of pairs: colluder to colluder (C-C), non-colluder to colluder (NC-C), non-colluder to non-colluder (NC-NC). The x-axis denotes the pairwise feature scores (collusiveness) and the y-axis denotes the corresponding frequencies.

collusiveness of three types of reviewer pairs, i.e., colluder to colluder pair, non-colluder to colluder pair, and non-colluder to non-colluder pair. Specifically, the collusiveness of (colluder,colluder) pairs should be higher than the other two. Figure 1 shows the distribution of each pairwise feature score (collusiveness) over the three types of reviewer pairs. Note that to be consistent with the way of collusiveness propagation in FraudInformer, only the pairs of reviewers whose positions in a product reviewer list are within an investigating range (IR) are considered. Here the IR is empirically set to be 25, close to the half of the average distance between two colluders in the dataset which is 56.07. We also compute the χ^2 value [19] for each feature; the larger the χ^2 value is, the higher discriminative power the corresponding feature has. From Figure 1 we can see that (colluder,colluder) pairs dominate the area of high collusiveness in each histogram. Among these pairwise features, BSD achieves the best performance not only in terms of the clear separation observed in the histogram but also the highest χ^2 values. Another notable observation is that brand-based pairwise features generally outperform the product-based counterparts (except PTD and BTD, however their difference is small), implying that colluders nowadays are more likely to target brands than specific products. By grouping different products with the same brands together, the collusive signals will be accumulated and become more prominent among colluders.

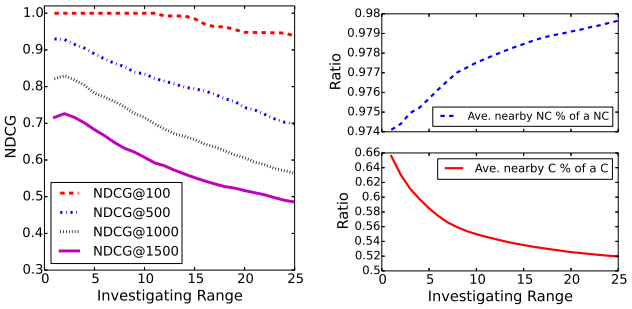
4.2 Effects of Investigating Range The optimal choice for investigating range is nontrivial; a small value may lead to some distant accomplices being not covered in the accusation (low recall) while a large one may not only bring along noise by accounting for some non-colluders (low precision), but may also result in inefficient computation. Figure 2 shows the effects of different investigating ranges (IRs) on the overall ranking performance. Here we model IR as a function of the length n_k of a product reviewer list ℓ_k (Section 2.1). We then consider four types of functions - Constant ($\zeta(n_k) = 5, 10, 15, 20$), Random ($\zeta(n_k) = random(1, 20)$), Root ($\zeta(n_k) = \sqrt{n_k}, \sqrt[3]{n_k}$), and Logarithmic ($\zeta(n_k) = \log_2(n_k + 1)$). From the experiments, we noted two observations.

The TYPE I noise. For constant IR functions, as IR increases from 5 to 20, we find a monotonic degradation on both NDCG and Precision (Figure 2). Further experiment (Figure 3(a)) is conducted with fine-grained IR$\in \{1, 2, ..., 25\}$ and the performance is shown to clearly decline as IR increases in higher ranks (k=100,500) and there is a small arch around IR=2 when k becomes larger (k=1000,1500). To derive further insights into why this happens, we compute the average ratio of nearby colluders in the data which turns out to decrease when IR rises (solid line in Figure 3(b))



(a) NDCG for top 1000 ranks (b) Precision for top 1000 ranks

Figure 2: Ranking performance for different investigating range (IR) settings. Uniform kernel function is used as the confidence function.



(a) NDCG vs. IRs (b) Neighborhood variations for different IRs

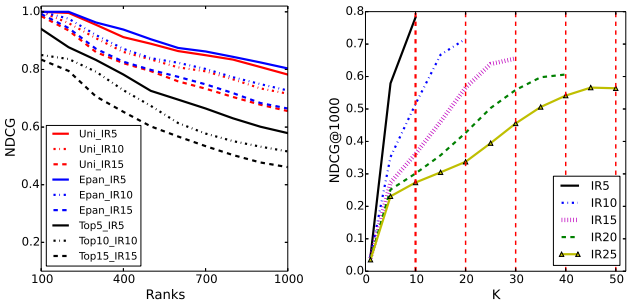
Figure 3: The rise of TYPE I noise from the collusion weight allocation in FraudInformer.

while the average ratio of nearby non-colluders increases slightly (dashed line in Figure 3(b)). This shows that as IR gets larger, in long enough product reviewer lists, a colluder will have more non-colluders involved in the accusation who aggregately share an increasing portion of collusion weights in the collusiveness propagation, leading to higher spamicities of these non-colluders in the final ranking (Eq.(2.6)). We denote this kind of noise rising from the collusion weight allocation in FraudInformer as the TYPE I noise which may lead to performance deterioration.

The TYPE II noise. We then consider the variable IRs (Random, Root, and Logarithmic) whose performance is much poorer than that of constant IRs. This is anti-intuitive because the performance of variable IRs is expected to remain moderate between those of larger constant IRs (e.g., 25) and smaller ones (e.g., 5) rather than worse than both cases. This may be partially due to the effect of the TYPE I noise because the IRs of longer product reviewer lists (ℓ_{Long}) are correspondingly larger, which makes it underperform small constant IRs. On the other hand, regarding the poorer performance compared to larger constant IRs, we argue that it may be attributed to two facts that in the dataset 1) most of the colluders write fake reviews for unpopular products with short reviewer lists (ℓ_{Short}), e.g., 51.5% colluders have reviewed 43.9% products with $|\ell_{Short}| \leq 12$ and nearly 3 of them (25%) are colluders on average, and 2) the positions of such colluders in ℓ_{Short} are quite close to each other, e.g., the average and median distances between two consecutive colluders in ℓ_{Short} with $|\ell_{Short}| \leq 12$ are 1.23 and 1 respectively. As such, when using the proportional variable IRs, in unpopular products a colluder will have less neighbors most of whom are also colluders with similarly high collusiveness, and in popular products a non-colluder will have more neighbors most of whom are also non-colluders with similarly low collusiveness. By normalizing the collusion weights to construct a stochastic transition matrix (Eq.(2.7)), the collusiveness in both cases will be smoothed out. Thus, non-colluders in ℓ_{Long} will gain higher "authority" during the global ranking due to the greater number of their non-colluder neighbors covered by larger IRs. In contrast, when using large constant IRs (e.g., 25), the colluders in ℓ_{Short} will have a portion of non-colluder neighbors as well, making the transition probability unevenly distributed over colluder neighbors and non-colluder ones which is just the proper reflection of the difference between the collusiveness value of C-C and C-NC pairs. We denote this kind of noise rising from the construction of transition matrix (Eq.(2.7)) as the TYPE II noise.

As a summary, a desirable investigating range should meet the following conditions: 1) it should not be set too large in *popular* products for restricting the "authority" of non-colluders and 2) it should not be set too small in *unpopular* products so as to maintain the coverage of colluders and also to boost the "authority" of colluders by providing the opportunity of intensifying the differences between positives (C-C pairs) and negatives (NC-C and NC-NC pairs) in the process of unsupervised learning.

4.3 Effects of Confidence Weighting We experiment with different confidence functions which encode different patterns of reviewers' confidence when providing evidence about each other. As shown in Figure 4, the Epanechnikov kernel with IR=5 (Epan_IR5) performs best and in terms of symmetric confidence the Epanechnikov kernel outperforms the uniform kernel in all IR settings. This is reasonable since the Epanechnikov kernel assigns more belief to the evidence about closer reviewers, which matches the concentration property of spam reviews [3]. On the other hand, the asymmetric function (the TopK kernel) is shown to outperform all symmetric counterparts. Recall that the TopK kernel trusts the evidence about the ones with higher collusiveness. To further analyze the nature of the TopK kernel, an additional experiment is conducted to study the impacts from different $K = \{1, 5, ..., 50\}$ and IR



(a) NDCG for top 1000 ranks (b) The performance of the TopK kernel (NDCG@1000)

Figure 4: Evaluation of different confidence functions.

settings (Figure 4(b)). Note that $K \leq 2\times$IR while $K = 2\times$IR is equivalent to the uniform kernel with IR=$K/2$ (the red dashed lines highlight the largest K corresponding to each IR setting). Several observations are noted. First, as K increases the performance gets better for all IR settings. This may be ascribed to the TYPE II noise; when K is small, for a specific colluder, the similarly high collusiveness with other nearby colluders will be smoothed out by the normalization, making it no difference with the case of non-colluders. Second, it shows that for a fixed K, the performance drops as IR becomes larger. This differs from the TYPE I noise as the number of neighbors does not change which is K all the time. A potential explanation is that although K is fixed, the top K selected neighbors for a particular reviewer are changing as IR becomes larger, and the chance of obtaining neighbors with similar collusiveness scores increases as well, which would amplify the effect of the TYPE II noise that the similarly high collusiveness of C-C pairs of a colluder would be neutralized after normalization. Finally, it shows that empirically the performance of the TopK kernel will converge to that of the uniform kernel as K goes to $2\times$IR.

4.4 Comparison Analysis We compare our framework with other competitors which exploit pointwise features for colluder detection.

(1) GSRank [11]: by modeling the relationships among spammers, spammer groups, and target products, GSRank, a state-of-the-art unsupervised algorithm for detecting review spammer groups, uses 8 well-designed group-based pointwise features to characterize colluder behaviors. We implement GSRank and the 8 group-based features for the comparison.

(2) Learning to Rank: this is another way of integrating the aforementioned pointwise group-based features where a training ranking can be obtained by sorting reviewers based on each feature in descending order (the group-based features assign higher scores to colluders than non-colluders). The resulting learned ranking model is in effect an optimal combination of the rankings produced by the 8 group-based feature functions. In our experiments, two learning to rank algorithms - SVMRank [6] and RankBoost [4] - are adopted and the default parameter settings are used.

For FraudInformer, according to previous experimental analysis, we choose the Epanechnikov kernel as the confidence function and the IR is set to 5. The algorithm converges after 150 iterations given that L_∞ norm is used to measure the difference between the spamicity vectors of consecutive iterations and the algorithm terminates when the difference $\leq 10^{-6}$. We compare with the learning to rank algorithms using 10-fold cross validation and the performance is averaged over all test folders (each test folder includes $\approx$ 500 examples). For GSRank, we use the entire dataset as one test folder since training is not needed in both GSRank and FraudInformer. All the improvements of our method over the baselines are significant at the confident level of 95% based on two-tailed t-test.

As shown in Table 1, our method in general outperforms both leaning to rank algorithms. For $k = 50$ the performance improves at least by 11.6% in Precision@k and 11.9% in NDCG@k. For comparison with GSRank[2], our method performs at least as well as GSRank. At higher rank positions (k=100,200,300) both methods can achieve promising results. However, when k exceeds 400, the performance of GSRank drops significantly. We further inspect the colluders ranked within [400, 1000] by FraudInformer yet missed by the top 1000 ranks of GSRank, finding 379 colluders in total of which 162 (42.7%) are ranked at the bottom 1000 by GSRank. Among the Bottom1000 colluders, 124 (124/162=76.5%) appear in ≤ 4 groups and we also find that some of the group behaviors are not well-captured by the proposed group based features (e.g., Group Size, Group Support Count, Group Deviation in [11]). This is reasonable since GSRank relies on the mutual enhancement between entities (i.e., groups, group members, and products); group members with low group feature scores will obtain low spamicities with GSRank. As a result, the inferred spamicities of smarter colluders who review many similar items with legitimate reviewers may be even lower than those active reviewers who happen to review many popular products. Moreover, among the 38 remaining GSRank Bottom1000 colluders, 24 of them are involved in big groups. This may also be problematic since larger groups may conceal some of the collusion signals exhibited by a portion of group members. In contrast, our method operates with a finer granularity which reveals connection between each pair of colluders directly. Although such colluders may not seem suspicious at group level, their collusiveness could be captured at pairwise level by the proposed pairwise features.

[2]The result of Precision@k is similar.

		SVMRank	RankBoost	FraudInformer
Precision@k	50	0.844	0.818	**0.942**
	100	0.804	0.780	**0.866**
	150	0.764	0.738	**0.785**
	200	**0.717**	0.701	0.708
	250	0.672	0.667	**0.688**
NDCG@k	50	0.853	0.825	**0.955**
	100	0.819	0.794	**0.890**
	150	0.784	0.759	**0.832**
	200	0.757	0.738	**0.773**
	250	0.830	0.820	**0.841**

Table 1: Performance comparison with learning to rank algorithms on both Precision and NDCG.

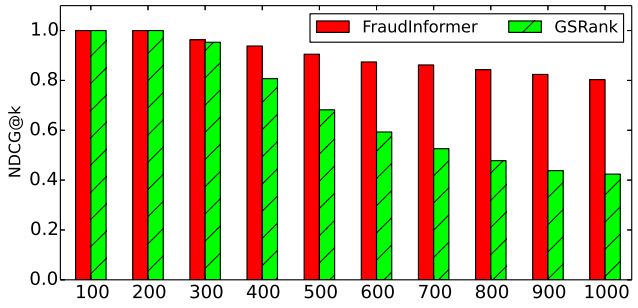


Figure 5: Performance comparison with GSRank on NDCG@k=100 to 1000.

5 The Adversarial Offense

In this section, we consider the adversarial challenge by answering the question that how much damage could an adversary cause when he tries to evade the detection of FraudInformer. By following the same rules and plans, colluders in a spam campaign are unlikely to have completely different targets or opposite opinions. However, they can manage their positions in the product reviewer lists so as to escape the coverage (investigating ranges) of other accomplices (e.g., if IR=10, colluders can place their spam reviews at an interval of 11). In the dataset we find that colluders are not always necessarily centralized within a specific region, and some may be far away from the majority. However, blindly improving the recall (i.e. to cover more colluders) by enlarging the IR may reduce the precision of FraudInformer (Section 4.2). To balance this trade-off, we have the following lemma.

Lemma 5.1. *Assume that in a spam campaign where each of N colluders is asked to write fake reviews for m out of M products which have reviewer lists with equal-length L. Each fake review generates one unit of impression which represents the utility of the spam. Then the average impression generated by this spam campaign on each product is bounded by $\frac{\gamma^{R-1}}{\gamma^R-1}$ if an exponential decay function $\frac{1}{\gamma^R}(\gamma>1)$ is used to model the impression where R is the investigating range.*

Proof. To evade the capture of other accomplices with investigating range R, the maximum impression will be achieved if the colluders are positioned with an interval of R, started from the top of each chronologically ordered reviewer list. Then the total impressions achieved by this spam campaign is:

$$\begin{aligned} I_{total} &= M\sum_{i=0}^{\lfloor \frac{mN}{M} \rfloor} \frac{1}{\gamma^{R\cdot i+1}} + (Nm - M\lfloor \frac{Nm}{M} \rfloor)\frac{1}{\gamma^{\lfloor \frac{Nm}{M} \rfloor+1}} \\ &\leq M\sum_{i=0}^{\lfloor \frac{mN}{M} \rfloor+1} \frac{1}{\gamma^{R\cdot i+1}} \leq M\frac{\frac{1}{\gamma}}{1-\frac{1}{\gamma_R}} = M\frac{\gamma^{R-1}}{\gamma^R-1} \end{aligned} \tag{5.18}$$

Then we have $I_{ave} = I_{total}/M = \frac{\gamma^{R-1}}{\gamma^R-1}$. □

We can see that the average impression decreases very fast, reciprocally, as the IR increases, e.g., when $\gamma = 1.1$, we have $\frac{I_{ave}(IR=1)}{I_{ave}(IR=10)} = 6.75$, which means that a moderate IR can significantly weaken and further limit the effects of spam campaigns. On the other hand, moderate IRs can effectively catch most of the colluders (Section 4.2) since the manipulation of review positions is not always an easy task in practice; for colluders it is unclear how long would it take for legitimate reviewers to post reviews right after their spam reviews being posted. Thus, to guarantee the achievement of predefined goals, colluders have to spam as quickly as possible, resulting in concentration in collusion attacks.

6 Related Work

Among all anti-spam approaches for online reviews, three detection tasks can be identified from the literature.

Spam Review Detection. In [5], duplicate and near duplicate reviews are assumed to be spam and are used to train a supervised model for detecting untruthful product reviews. Li et al. [7] use reviewer related information (e.g., user profiles, brand) to help detect review spam by incorporating review- and reviewer-level features into a semi-supervised model. In [12], much attention is paid to review text. Linguistic and psycholinguistic features are combined to train a highly accurate classifier for spam review detection. However, review text is vulnerable to manipulation and it has also been shown that humans are poor at identifying deceptive reviews by just reading the review text [12].

Review Spammer Detection. In [16], the causality among reviews, reviewers, and stores has been used to construct a heterogeneous graph for spammer detection. A more principled framework proposed in [1] is based on Markov random field (MRF) where a signed bipartite review network is created to link reviewers and products (nodes) with reviews (edges). In addition, some studies are focused on spamming behavior analysis. Lim et al. [8] propose multiple behavioral features to model the potential spamming practices for reviewers. Mukherjee

et al. [10] integrate state-of-the-art behavioral features into an unsupervised Bayesian framework where the spamicity of a reviewer is modeled as a latent variable. Although shown to be effective to detect spammers, these approaches may be problematic when confronting with colluders since collective behaviors of colluders may not appear suspicious if the targets are attacked by a large number of well-organized colluders who do not appear as outliers any more.

Group Spammer Detection. Our work belongs to this direction which has not received much attention so far. In [11], several group based pointwise features are proposed to capture collective behaviors of candidate reviewer groups. A ranking algorithm is then used to rank these candidates based on these features. Our study has several distinctions. First, it does not rely on the concept of "group" which may have granularity problem; a tiny group may not be able to exhibit sufficiently suspicious collective behaviors so as to be captured by the group based features. Second, our proposed pairwise features are capable to directly reveal the intrinsic connections among colluders whereas the group based features are restricted by the group itself in that each measurement has to be aligned to the group-level, so that the collusion evidence about portions of group members may be concealed by the behaviors of the whole group. Xu et al. [18] spot colluders by building a supervised MRF model based on co-reviewing behaviors of reviewers. In contrast, our method works with no prior knowledge, which is more desirable as obtaining the ground truth about review spam is usually very hard and unreliable in practice.

7 Conclusions

In this paper, we conduct study on the detection of colluders in online review spam campaigns via multiple heterogeneous pairwise features. We find that pairwise features can directly and fine-grainedly model the behavioral proximity of colluders, e.g., reviewing similar products and expressing similar opinions within short time periods. We also propose an intuitive framework utilizing accompanied benefits from pairwise features upon an autonomous process that makes colluders themselves give each other away, which is also scalable and unsupervised. Empirical experiments validate the effectiveness of our method and show that it significantly outperforms baselines by revealing colluders who have managed to evade the capture of the state-of-the-art pointwise group spammer features.

Acknowledgements

We thank the anonymous reviewers for their helpful comments. This work is supported by the MOE AcRF Tier 2 Grant M4020110.020 awarded to Dr. Jie Zhang.

References

[1] Leman Akoglu, Rishi Chandy, and Christos Faloutsos, *Opinion fraud detection in online reviews by network effects*, in Proceedings of ICWSM, 2013.

[2] C. Chen, K. Wu, V. Srinivasan, and X. Zhang, *Battling the internet water army: Detection of hidden paid posters*, CoRR arXiv:1111.4297, (2011).

[3] Geli Fei, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh, *Exploiting burstiness in reviews for review spammer detection*, in Proceedings of ICWSM, 2013.

[4] Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer, *An efficient boosting algorithm for combining preferences*, JMLR, 4 (2003), pp. 933–969.

[5] N. Jindal and B. Liu, *Opinion spam and analysis*, in Proceedings of WSDM, 2008.

[6] Thorsten Joachims, *Optimizing search engines using clickthrough data*, in Proceedings of KDD, 2002.

[7] F. Li, M. Huang, Y. Yang, and X. Zhu, *Learning to identify review spam*, in Proceedings of IJCAI, 2011.

[8] E.P. Lim, V.A. Nguyen, N. Jindal, B. Liu, and H.W. Lauw, *Detecting product review spammers using rating behaviors*, in Proceedings of CIKM, 2010.

[9] Michael Luca and Georgios Zervas, *Fake it till you make it: Reputation, competition, and yelp review fraud*, Harv. Business School NOM Unit Work. Paper, (2013).

[10] Arjun Mukherjee, Abhinav Kumar, Bing Liu, Junhui Wang, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh, *Spotting opinion spammers using behavioral footprints*, in Proc. of KDD, 2013.

[11] A. Mukherjee, B. Liu, and N. Glance, *Spotting fake reviewer groups in consumer reviews*, in Proc. of WWW, 2012.

[12] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock, *Finding deceptive opinion spam by any stretch of the imagination*, arXiv preprint arXiv:1107.4557, (2011).

[13] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd, *The pagerank citation ranking: bringing order to the web.*, tech. report, Stanford University, Stanford, CA, 1998.

[14] Emanuel Parzen, *On estimation of a probability density function and mode*, The annals of mathematical statistics, 33 (1962).

[15] David Streitfeld, *The best book reviews money can buy.* www.nytimes.com/2012/08/26/business/book-reviewers-for-hire-meet-a-demand-for-online-raves.html.

[16] G. Wang, S. Xie, B. Liu, and S.Y. Philip, *Identify online store review spammers via social review graph*, ACM TIST, 3 (2012).

[17] Sihong Xie, Guan Wang, Shuyang Lin, and Philip S Yu, *Review spam detection via temporal pattern discovery*, in Proceedings of KDD, 2012.

[18] Chang Xu, Jie Zhang, Kuiyu Chang, and Chong Long, *Uncovering collusive spammers in chinese review websites*, in Proceedings of CIKM, 2013.

[19] Yiming Yang and Jan O Pedersen, *A comparative study on feature selection in text categorization*, in Proceedings of ICML, 1997.