Towards Collusive Fraud Detection in Online Reviews

Chang Xu
School of Computer Engineering
Nanyang Technological University, Singapore
xuch0007@e.ntu.edu.sg

Jie Zhang
School of Computer Engineering
Nanyang Technological University, Singapore
zhangj@ntu.edu.sg

***Abstract*—Online review fraud has evolved in sophistication by launching intelligent campaigns where a group of coordinated participants work together to deliver deceptive reviews for the designated targets. Such collusive fraud is considered much harder to defend against as these campaign participants are capable of evading detection by shaping their behaviors collectively so as not to appear suspicious. The present work complements existing studies by exploring more subtle behavioral trails connected with collusive review fraud. A novel statistical model is proposed to further characterize, recognize, and forecast collusive fraud in online reviews. The proposed model is completely unsupervised, which bypasses the difficulty of manual annotation required for supervised modeling. It is also highly flexible to incorporate collusion characteristics available for better modeling and prediction. Experiments on two real-world datasets demonstrate the effectiveness of the proposed method and the improvements in learning and predictive abilities.**

***Keywords*-Collusive Review Fraud; Opinion Spam**

I. Introduction

As online reviews have become increasingly influential in helping online shoppers make purchase decisions, review fraud [1] has emerged as a major threat to this process. This blackhat practice intends to affect people's buying decisions by creating misleading reviews about particular businesses (e.g., restaurants, hotels). By committing review fraud, malicious business owners can often achieve sales increase by posting false positive reviews for themselves or leaving false negative reviews for their rivals. It has been estimated that about 16% of Yelp reviews written for the restaurants in the metropolitan Boston area are fake [2]. To make the situation even worse, review fraud practitioners today have evolved in specialization; they were found to collaborate and form coordinated campaigns [3], [4] such that richer manpower and trickier tactics can be put into use to achieve more covert and cost-effective fraud practices.

There are prior attempts at tackling such *collusive fraud*. Supervised approaches were proposed for detecting review fraud campaigns [5], [4]. Although shown to possess high accuracy, these methods rely heavily on real fake reviews for model training, which is highly challenging due to the lack of ground truth in real-world scenarios [6]. To overcome this problem, [3] pioneered the exploration of a variety of *collective behaviors* of reviewers, and proposed an unsupervised model to rank reviewers in an iterative manner. [7] studied the network-based characteristics of spammers under the same campaigns and clustered these spammers based on their commonly reviewed products. Nevertheless, a major problem is that these approaches cannot learn from existing data for making predictions about emerging collusion. Predictive models are crucial to fraud detection tasks as they are deployable in real-time scenarios where emerging fraud practices should be timely detected and removed so as to minimize the caused damages. Moreover, being a deterministic model by nature, these approaches could be sensitive to the variability in the feature data that can be incurred for different reasons, such as particular parameter settings and noisy input data.

In this work, we are motivated to fill those gaps by modeling collusive fraud[1] in online reviews. In particular, we identify the problem of detecting collusive review fraud from a stochastic perspective, and seek for a statistical solution that can 1) infer occurred collusion in existing unlabeled data as unsupervised models, 2) learn to make collusion forecasts as supervised models, and 3) handle uncertainty in the measurements of used features. In the light of such objectives, we propose the Latent Collusion Model (LCM), a novel statistical model that fulfils all the above goals. The key perspective of LCM is based on the appreciation of revealed characteristics that can differentiate colluders from non-colluders in specific feature space (e.g., [3], [4]). Specifically, colluders have been found to exhibit *unique collective behavior patterns* that result from their collaborations. It is then possible for LCM to build a unified model for both tasks of collusion inference and prediction by taking a hybrid generative and discriminative probabilistic approach.

Moreover, to complement existing collusion-oriented features and expend the feature space for characterizing collusive review fraud, we propose a suite of *homogeneity-based* collusive behavior measures (h-CBMs) to distinguish colluders from non-colluders. h-CBMs focus on the intrinsic connections between colluders by measuring the similarity between the behaviors of a group of related

[1]Throughout this paper, we will use the term "colluder" to refer to those who have collaborated with each other in any collusive fraud attack.

Variable	Description
v; $\mathcal{V}$	A reviewer v; the set of all reviewers
g; $\mathcal{M}_g$	A candidate colluder group; the members of g
$\mathcal{I}_v$; $\mathcal{I}_g$	Businesses reviewed by v; Businesses reviewed by g
$r_{v,i}$	Rating given to business i by v
$t_{v,i}$	Timestamp of the review given by v to business i

Table I: Notations for defining h-CBMs.

colluders. Through h-CBMs, we find that to complete the tasks assigned by a campaign, colluders tend to take very similar actions such as targeting almost the same businesses, providing highly consistent review ratings, and reviewing at proximate times. To avoid detection, even though colluders can restrict the size of launched campaigns or overwhelm their targets by review flooding, the operations they need to finish their tasks are inevitably homogeneous, which renders them detectable.

II. Characterizing Collusive Review Fraud

As discussed, colluders under the same campaigns can exhibit unique collective behavior patterns. However, existing features for this are mainly designed for finding large campaigns with a non-trivial number of colluders and targets (e.g., *Group Size* and *Group Support Count* in [3]); they are likely to miss stealthier campaigns with smaller sizes. Also, there are features regarding colluders as anomaly and measuring how much their behaviors would deviate from those of others (e.g., *Group Deviation* and *Group Size Ratio* in [3]). However, colluders could easily turn themselves into the majority by review-flooding their targets so that the deviations can be eliminated. To cope with these issues, we develop the homogeneity-based collusive behavior measures (h-CBMs) that inspect the homogeneity in collective behaviors of colluders, i.e., the similar behaviors exhibited during their working for the same campaigns.

In particular, to find reviewers who are likely to exhibit collective behaviors, we borrow the concept of *candidate colluder group*[2] from [3] to refer to a group of reviewers who have co-reviewed multiple businesses. The authors use frequent itemset mining (FIM) to generate groups consisting of at least two reviewers who have co-reviewed at least three businesses. In this paper, h-CBMs will also act on these groups. For terminology, a candidate colluder group, or *group* for short, is regarded as *malicious* and its members as *colluders* if its members have exhibited collective behaviors related to collusive fraud. Otherwise, it is *benign*. Next, we introduce the h-CBMs for characterizing collusive review fraud. Tables I and II give the notations and definitions respectively.

(1) **Target-based h-CBMs**. As no campaigns can launch without specifying the targets, the connections between colluders can be best revealed by inspecting the businesses they focus on. Colluders working for the same campaigns will be assigned to the same targets. If the members of a group possess highly consistent reviewing histories, they are very likely to be involved in collusive fraud (*Target Consistency*).

[2]In [3], the term used is *candidate spammer group*.

Name	Description
Target Consistency (TC)	Jaccard Similarity of business sets reviewed by members of g: $\#\{\cap_{v\in\mathcal{M}_g}\mathcal{I}_v\}/\#\{\cup_{v\in\mathcal{M}_g}\mathcal{I}_v\}$
Rating Consistency (RC)	Max. variance of review ratings given by $\mathcal{M}_g$ to businesses in $\mathcal{I}_g$: $\max_{i\in\mathcal{I}_g}(\mathrm{var}(\{r_{v,i}\|v\in\mathcal{M}_g\}))$
Temporal Sync. (TS)	Max. variance of timestamps of reviews posted by $\mathcal{M}_g$ to businesses in $\mathcal{I}_g$: $\max_{i\in\mathcal{I}_g}(\mathrm{var}(\{t_{v,i}\|v\in\mathcal{M}_g\}))$
First-review Sync. (FS)	Variance of timestamps of the first reviews posted by members in g: $\mathrm{var}(\{\min(\{t_{v,i}\|i\in\mathcal{I}_v\})\|v\in\mathcal{M}_g\})$
Activity Consistency (AC)	Variance of the most active moments of members of g. The most active moment of a reviewer is the date on which (s)he posts the most reviews: $\mathrm{var}(\{t_v^{max}\|v\in\mathcal{M}_g\})$, $t_v^{max}=\arg\max_{t\in T_v}(\#\{t_{v,i}=t\|i\in\mathcal{I}_v\})$
Workload Sim. (WS)	Variance of the numbers of businesses reviewed by members in g: $\mathrm{var}(\{\|\mathcal{I}_v\|\|v\in\mathcal{M}_g\})$

Table II: Summary of the h-CBMs.

(2) **Rating-based h-CBMs**. As a major vehicle for expressing opinions, review ratings may also be subjected to manipulation. Colluders can easily dominate the overall opinions about the targets by creating a large number of high ratings for promotion or low ratings for vilification. If a group often gives consistently high/low ratings to the rated businesses, it is likely to be suspicious (*Rating Consistency*).

(3) **Temporal-based h-CBMs**. Timing is vital to fraud campaign services as efficiency usually brings more profits. Colluders are often required to finish their tasks in time, leading to their synchronized behaviors being observed [8]. For this, we compute the variance of the reviewing timestamps of the members of a group (*Temporal Sync.*). As colluders tend to use accounts auto-generated or bought in batch [9], it is possible to spot another kind of synchronicity by comparing the times when colluders use their accounts for the first time (*First-review Sync.*).

(4) **Activity-based h-CBMs**. Having to follow the same campaign schedules and share the overall reviewing workload, colluders may exhibit very similar activeness patterns. For the campaign schedule factor, we measure the similarity between the most active moments of the members of a group (*Activity Consistency*). For the workload sharing factor, one way to measure the workload would be counting the total number of reviews posted by a reviewer. To share the overall workload of a campaign, the workload assigned to each colluder may be similar (*Workload Sim.*).

A merit of h-CBMs is that they are parameter-free, which omits the need for parameter estimation based on held-out labeled data. Numerically, each h-CBM is standardized using 0-1 scaling and thus has value in [0,1]. The variance-based h-CBMs (all except TC) are converted to similarity-based measures by using the formula: $s_{\text{new}}=\frac{2}{1+s_{\text{old}}}-1$; a value closer to 1 suggests a group be more likely to be malicious.

III. Latent Collusion Model

In this section, we introduce the Latent Collusion Model (LCM) for modeling collusive review fraud from a probabilistic view. Given a collection of candidate colluder groups and their h-CBM measurements, LCM aims, in an unsupervised way, to infer the occurred collusion and meanwhile build a collusion predictor for making forecasts. A principled optimization algorithm is also implemented to conduct model learning and inference.

A. Problem Formulation

We are given a set of N candidate colluder groups $\boldsymbol{\mathcal{G}} = \{g_1, \ldots, g_N\}$ in a review site, in which each group g_n is associated with an M-dimensional h-CBM[3] vector $\boldsymbol{\phi}_n = \{\phi_n^{(1)}, \ldots, \phi_n^{(M)}\}$, $\phi_n^{(m)} \in [0, 1]$. Then the set of all h-CBM vectors $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N\}$ constitutes our observed data about all the groups $\boldsymbol{\mathcal{G}}$. The class label of each group g_n is denoted by a binary variable $z_n \in \{0, 1\}$, specifying whether it is benign or malicious. As our context is unsupervised, all the class labels $\boldsymbol{Z} = \{z_1, \ldots, z_N\}$ of $\boldsymbol{\mathcal{G}}$ are unknown.

Now we define the problem of detecting collusive review fraud from a *probabilistic modeling* perspective as follows:

Problem 1: **Given** a set of candidate colluder groups $\boldsymbol{\mathcal{G}}$ with their observed h-CBM vectors $\boldsymbol{\Phi}$, the problem of detecting collusive fraud in online reviews involves two subtasks: 1) **infer** the posterior distribution of the class labels $\boldsymbol{Z}$ of $\boldsymbol{\mathcal{G}}$, $p(\boldsymbol{Z}|\boldsymbol{\Phi})$, and 2) **predict** the class label $\hat{z}$ for an emerging group $\hat{g}$ based on the predictive distribution $p(\hat{z}|\hat{\boldsymbol{\phi}}, \boldsymbol{\Phi}, \boldsymbol{Z})$, where $\hat{\boldsymbol{\phi}}$ is the h-CBM vector of $\hat{g}$.

B. The Model

To solve Problem 1, for the collusion inference task, we need to compute the posterior distribution $p(\boldsymbol{Z}|\boldsymbol{\Phi})$, while for the collusion prediction task, we need to derive the predictive distribution $p(\hat{z}|\hat{\boldsymbol{\phi}}, \boldsymbol{\Phi}, \boldsymbol{Z})$. It can be seen that in both cases the unknown quantity $\boldsymbol{Z}$ plays an important role, and can be used to connect the two parts in a way that one can benefit from the other. For this, LCM treats the class labels $\boldsymbol{Z}$ as latent, and considers the *reciprocal* relationship between the unknown class labels $\boldsymbol{Z}$ and the observed h-CBM vectors $\boldsymbol{\Phi}$ from two probabilistic modeling views: generative and discriminative.

The generative view: occurred collusion inference. The generative view of LCM considers the class labels as the *hidden factor* that causes the h-CBM vectors. As mentioned, colluders often possess unique collective behavior patterns in feature space. Once we know the class label of a group, we can somehow generate its h-CBM vector based on the difference between the collective behavior patterns (distributions) of malicious and benign groups, which essentially yields two distinctive clusters. Then we can take a clustering-based generative approach [10] to infer the posterior distribution $p(\boldsymbol{Z}|\boldsymbol{\Phi})$ so as to solve the inference task of Problem 1.

Specifically, the generative approach for this clustering postulates a generative process describing how the h-CBM vectors of a mixture of malicious and benign groups can be generated by LCM: for each group g_n, to generate its M-dimensional h-CBM vector ϕ_n, we would first decide its class label z_n based on a cluster membership distribution $p(z_n)$, and then for each h-CBM dimension $\phi_n^{(m)}$, we would draw a value from the cluster-conditional distribution parametrized as $p(\phi_n^{(m)}|\boldsymbol{\lambda}_{z_n}^{(m)})$. The parameter $\boldsymbol{\lambda}_{z_n}^{(m)} = \{\boldsymbol{\lambda}_0^{(m)}, \boldsymbol{\lambda}_1^{(m)}\}$ essentially captures the latent collective behavior patterns of colluders ($\boldsymbol{\lambda}_1^{(m)}$) and non-colluders ($\boldsymbol{\lambda}_0^{(m)}$) on the mth h-CBM dimension. As each h-CBM dimension $\phi_n^{(m)}$ takes values in $[0, 1]$, we substantialize the cluster-conditional distribution $p(\phi_n^{(m)}|\boldsymbol{\lambda}_{z_n}^{(m)})$ as a standard Beta distribution $\mathbf{Beta}(\phi_n^{(m)}|\alpha_{z_n}^{(m)}, \beta_{z_n}^{(m)})$ with parameters $\alpha_{z_n}^{(m)}$ and $\beta_{z_n}^{(m)}$, such that $\boldsymbol{\lambda}_{z_n}^{(m)} = [\alpha_{z_n}^{(m)}, \beta_{z_n}^{(m)}]^T$.

Given this generative process, the likelihood of generating the observed data $\boldsymbol{\Phi}$ based on the model parameter $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_0^{(m)}\}_{m=1}^M \cup \{\boldsymbol{\lambda}_1^{(m)}\}_{m=1}^M$ can be written as:

$$\mathcal{L}(\boldsymbol{\lambda}|\boldsymbol{\Phi}, \boldsymbol{Z}) = \prod_{n=1}^{N} \prod_{m=1}^{M} \sum_{k \in \{0,1\}} p(z_n = k) \cdot \mathbf{Beta}(\phi_n^{(m)}|\alpha_{z_n}^{(m)}, \beta_{z_n}^{(m)}) \tag{1}$$

Then we are able to compute the posterior distribution $p(\boldsymbol{Z}|\boldsymbol{\Phi}, \boldsymbol{\lambda})$ by maximizing the likelihood in Eq. (1) with respect to the model parameter $\boldsymbol{\lambda}$ with an EM algorithm [11]. More details will be presented later in Section III-C.

The discriminative view: emerging collusion prediction. The discriminative view, on the other hand, regards the class labels as the *hidden consequence* of the observed h-CBM vectors. This view is considered more natural since the very goal of h-CBMs is to discriminate between colluders and non-colluders; given the h-CBM vector of a group, we can tell its class label by referring to the semantics of h-CBMs.

To solve the prediction task of Problem 1, i.e., to compute the predictive distribution $p(\hat{z}|\hat{\boldsymbol{\phi}}, \boldsymbol{\Phi}, \boldsymbol{Z})$, we take a discriminative approach [10] where $p(z_n|\boldsymbol{\phi}_n)$ is directly defined. Here, we use a logistic function to model this distribution: $p(z_n = 1|\boldsymbol{\phi}_n, \mathbf{w}) = \sigma(\mathbf{w}^T \boldsymbol{\phi}_n)$, where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the logistic function, $\mathbf{w} = \{\mathrm{w}_0, \cdots, \mathrm{w}_M\}$ is an $(M+1)$-dimensional weight vector to be estimated; w_0 is the weight for an additional dummy h-CBM $\boldsymbol{\phi}_n^{(0)}$=1 which is added for notational compactness. In general, other suitable models for binary responses could also apply, such as the probit regression model [12]. Then, predictions for emerging groups can be made by marginalizing the predictive distribution with respect to w:

$$p(\hat{z} = 1|\hat{\boldsymbol{\phi}}, \boldsymbol{\Phi}, \boldsymbol{Z}) = \int p(\hat{z} = 1|\hat{\boldsymbol{\phi}}, \mathbf{w}) p(\mathbf{w}|\boldsymbol{\Phi}, \boldsymbol{Z}) \mathrm{d}\mathbf{w} \tag{2}$$

However, this marginalization requires the knowledge of all class labels $\boldsymbol{Z}$, and as our context is unsupervised, we are

[3]Note that other collusion-oriented features can also be used in LCM. The term "h-CBM" here is for representational purpose only.

not provided with such information.

The full model. Indeed, as we have already seen the ability of LCM to perform collusion inference before, we can instead make use of the class labels inferred *concurrently* from the generative process of LCM as the pseudo-ground truth to derive the predictive distribution using Eq. (2). For this, we need to combine the generative and discriminative views of LCM to produce the final model. Specifically, the two views combine in a way that the joint probability of all the h-CBM vectors $\mathbf{\Phi}$ and the class labels $\mathbf{Z}$ given the model parameters $\{\mathbf{w}, \boldsymbol{\lambda}\}$ is written as:

$$
\begin{aligned}
p(\mathbf{\Phi}, \mathbf{Z}|\mathbf{w}, \boldsymbol{\lambda}) &= \prod_{n=1}^{N} p(z_n|\boldsymbol{\phi}_n, \mathbf{w}) \prod_{m=1}^{M} p(\phi_n^{(m)}|\boldsymbol{\lambda}_{z_n}^{(m)}) \\
&= \prod_{n=1}^{N} \left[\frac{1}{1+\exp(-\mathbf{w}^T\boldsymbol{\phi}_n)}\right]^{z_n} \left[\frac{\exp(-\mathbf{w}^T\boldsymbol{\phi}_n)}{1+\exp(-\mathbf{w}^T\boldsymbol{\phi}_n)}\right]^{1-z_n} \\
&\cdot \Bigg(\prod_{m=1}^{M} \left[\frac{\Gamma(\alpha_1^{(m)}+\beta_1^{(m)})}{\Gamma(\alpha_1^{(m)})\Gamma(\beta_1^{(m)})} (\phi_n^{(m)})^{\alpha_1^{(m)}-1}(1-\phi_n^{(m)})^{\beta_1^{(m)}-1}\right]^{z_n} \\
&\cdot \left[\frac{\Gamma(\alpha_0^{(m)}+\beta_0^{(m)})}{\Gamma(\alpha_0^{(m)})\Gamma(\beta_0^{(m)})} (\phi_n^{(m)})^{\alpha_0^{(m)}-1}(1-\phi_n^{(m)})^{\beta_0^{(m)}-1}\right]^{1-z_n} \Bigg)
\end{aligned} \tag{3}
$$

C. Learning and Inference

To apply the proposed LCM for modeling collusive review fraud, we need to compute the posterior distribution of latent variables and estimate the model parameters. In LCM, the latent variables $\mathbf{Z} = \{z_n\}_{n=1}^N$ account for the occurrence of collusion in the input data. The model parameters are $\{\mathbf{w}, \boldsymbol{\lambda}\}$ where $\mathbf{w}$ enables the learning of a collusion predictor, and $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_0^{(m)}\}_{m=1}^M \cup \{\boldsymbol{\lambda}_1^{(m)}\}_{m=1}^M$ captures collective behavior patterns of colluders and non-colluders. As LCM is a latent variable model, we develop an EM algorithm to perform model learning and inference for LCM.

E-step: We derive the posterior of latent variable $\mathbf{Z}$ based on Bayes' theorem and Eq. (3) as follows:

$$
\begin{aligned}
q(\mathbf{Z}|\mathbf{\Phi}, \mathbf{w}, \boldsymbol{\lambda}) &\propto p(\mathbf{\Phi}, \mathbf{Z}|\mathbf{w}, \boldsymbol{\lambda}) \\
&= \prod_{n=1}^{N} \left[\sigma(\mathbf{w}^T\boldsymbol{\phi}_n) \prod_{m=1}^{M} \mathbf{Beta}(\phi_n^{(m)}|\alpha_1^{(m)}, \beta_1^{(m)})\right]^{z_n} \\
&\cdot \left[(1-\sigma(\mathbf{w}^T\boldsymbol{\phi}_n)) \prod_{m=1}^{M} \mathbf{Beta}(\phi_n^{(m)}|\alpha_0^{(m)}, \beta_0^{(m)})\right]^{1-z_n}
\end{aligned} \tag{4}
$$

For simplicity, it is assumed that each group is independent with each other and so is its label. Then, by factorizing Eq. (4), we obtain the posterior of each z_n as:

$$
q(z_n = 1|\boldsymbol{\phi}_n, \mathbf{w}, \boldsymbol{\lambda}) = \sigma(\mathbf{w}^T\boldsymbol{\phi}_n + \triangle_{\boldsymbol{\lambda}}) \tag{5}
$$

with $\triangle_{\boldsymbol{\lambda}}$ defined as:

$$
\begin{aligned}
\triangle_{\boldsymbol{\lambda}} = &\sum_{m=1}^{M} \log\Gamma(\alpha_1^{(m)}+\beta_1^{(m)}) + \log\Gamma(\alpha_0^{(m)}) + \log\Gamma(\beta_0^{(m)}) \\
&- \log\Gamma(\alpha_0^{(m)}+\beta_0^{(m)}) - \log\Gamma(\alpha_1^{(m)}) - \log\Gamma(\beta_1^{(m)}) \\
&+ (\alpha_1^{(m)} - \alpha_0^{(m)}) \log\phi_n^{(m)} + (\beta_1^{(m)} - \beta_0^{(m)}) \log(1-\phi_n^{(m)})
\end{aligned}
$$

M-step: We find the estimate for model parameters $\{\mathbf{w}, \boldsymbol{\lambda}\}$ so as to maximize the expected value of the complete data log-likelihood $\mathbb{E}_{\boldsymbol{Z}}[\log p(\mathbf{\Phi}, \mathbf{Z}|\mathbf{w}, \boldsymbol{\lambda})]$.

For $\mathbf{w}$, due to the nonlinearity of logistic function, the optimal solution cannot be found analytically. We appeal to the Newton-Raphson method where the optimal $\mathbf{w}^*$ can be estimated iteratively:

$$
\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} - \mathbf{H}_{\mathbf{w}}^{-1} g_{\mathbf{w}} \tag{6}
$$

with gradient $g_{\mathbf{w}}$=$\mathbf{\Phi}^T(\boldsymbol{q}^{\bar{z}} - \boldsymbol{p}^{\bar{z}})$ and Hessian $\mathbf{H}_{\mathbf{w}}$=$\mathbf{\Phi}^T\boldsymbol{Q}\mathbf{\Phi}$, where $\boldsymbol{q}^{\bar{z}}$ is a column vector $\{\mathbb{E}_q[z_n]\}_{n=1}^N$; $\boldsymbol{p}^{\bar{z}}$ a column vector $\{\mathbb{E}_p[z_n]\}_{n=1}^N$; $\boldsymbol{Q}$ an $N \times N$ diagonal matrix with elements $Q_{nn} = \mathbb{E}_q[z_n](1 - \mathbb{E}_q[z_n])$.

For $\boldsymbol{\lambda} = \{\alpha_k^{(m)}, \beta_k^{(m)}\}_{m=1}^M$, $k \in \{0, 1\}$, the optimal value $\{\alpha_k^{(m)^*}, \beta_k^{(m)^*}\}$ for each parameter pair are coupled and need to be solved simultaneously. Again, we resort to the Newton-Raphson method where each Beta parameter pair $\lambda_k^{(m)} = \{\alpha_k^{(m)}, \beta_k^{(m)}\}$ can be optimized iteratively:

$$
\lambda_k^{(m)^{\text{new}}} = \lambda_k^{(m)^{\text{old}}} - \mathbf{H}_{\lambda_k}^{-1} \mathbf{g}_{\lambda_k} \tag{7}
$$

where gradients $\mathbf{g}_{\lambda_k} = [g_{\lambda_k}^{\alpha}, g_{\lambda_k}^{\beta}]$:

$$
g_{\lambda_k}^{\alpha} = \sum_n q(z_n = k)\{\log\phi_n^{(m)} - [\psi(\alpha_k^{(m)}) - \psi(\alpha_k^{(m)} + \beta_k^{(m)})]\}
$$

$$
g_{\lambda_k}^{\beta} = \sum_n q(z_n = k)\{\log(1-\phi_n^{(m)}) - \psi(\beta_k^{(m)}) - \psi(\alpha_k^{(m)} + \beta_k^{(m)})]\}
$$

and Hessian $\mathbf{H}_{\lambda_k}$=$\sum_n q(z_n = k) \cdot \mathbf{H}_0$ with $\mathbf{H}_0$ defined as:

$$
\mathbf{H}_0 = \begin{bmatrix} \psi'(\alpha_k^{(m)} + \beta_k^{(m)}) - \psi'(\alpha_k^{(m)}) & \psi'(\alpha_k^{(m)} + \beta_k^{(m)}) \\ \psi'(\alpha_k^{(m)} + \beta_k^{(m)}) & \psi'(\alpha_k^{(m)} + \beta_k^{(m)}) - \psi'(\beta_k^{(m)}) \end{bmatrix}
$$

where $\psi(\cdot)$ is the *digamma* function; $\psi'(\cdot)$ the *trigamma* function.

Finally, the full EM algorithm proceeds as follows. First, we initialize $\{\boldsymbol{\lambda}_k^{(m)}\}$ with uninformative prior $[1, 1]$, and $\boldsymbol{w}$ randomly. Then the algorithm is performed iteratively by alternatively executing the *E-step* and *M-step* in each iteration until the log-likelihood $\log p(\mathbf{\Phi}, \mathbf{Z}|\mathbf{w}, \boldsymbol{\lambda})$ converges.

D. Collusion Inference and Prediction

After performing model learning and inference for LCM, we can finally solve Problem 1.

For the collusion inference task, one can compute the probability of each group g_n being malicious by referring to the posterior of its class label z_n in Eq. (5):

$$
q(z_n = 1|\boldsymbol{\phi}_n, \mathbf{w}^*, \boldsymbol{\lambda}^*) = \sigma(\mathbf{w}^{*T}\boldsymbol{\phi}_n + \triangle_{\boldsymbol{\lambda}^*}) \tag{8}
$$

where $\mathbf{w}^*$ and $\boldsymbol{\lambda}^*$ are the (local) optimal solution obtained after the EM algorithm converges.

For the collusion prediction task, based on $\mathbf{w}^*$ and Eq. (2), the collusion predictive distribution can be derived as:

$$
\begin{aligned}
p(\hat{z} = 1|\hat{\boldsymbol{\phi}}, \mathbf{\Phi}, \boldsymbol{Z}) &= \int p(\hat{z} = 1|\hat{\boldsymbol{\phi}}, \mathbf{w}) p(\mathbf{w}|\mathbf{\Phi}, \boldsymbol{Z}) \mathrm{d}\mathbf{w} \\
&\approx \int p(\hat{z} = 1|\hat{\boldsymbol{\phi}}, \mathbf{w}^*) p(\mathbf{w}|\mathbf{\Phi}, \boldsymbol{Z}) \mathrm{d}\mathbf{w} \\
&= p(\hat{z} = 1|\hat{\boldsymbol{\phi}}, \mathbf{w}^*) = \sigma(\mathbf{w}^{*T}\boldsymbol{\phi}_n)
\end{aligned} \tag{9}
$$

IV. Experiments

We now evaluate the proposed LCM and conduct comparison analysis with several baselines based on two real-world consumer review datasets.

A. Datasets

Our evaluation is conducted on two real-world consumer review datasets that contain collusive fraud. The first one (denoted by "DL") has been used in [4] which contains 1,205,125 reviews posted by 645,072 reviewers for 136,785 products on Amazon.cn. Among DL, a total of 8,915 groups have been identified (involving 1,937 colluders and 3,118 non-colluders). The second dataset (denoted by "DU") has been used in [3], with 7,052 groups obtained from a set of 109,518 consumer reviews posted by 53,469 reviewers for 39,392 products on Amazon.com. As we cannot get access to the labels of groups in DU, we adopt an unsupervised evaluation method to experiment with DU.

B. Performance Comparison on DL

In this section, we compare the predictive performance of LCM to other unsupervised approaches using the annotated DL. For evaluation purpose, two well-known ranking based metrics, namely Average Precision (AP) and Area Under ROC Curve (AUC) are used.

Baselines. The problem studied in [3] is most closely related to ours where an unsupervised ranking model, *GSRank*, is proposed to find malicious reviewer groups. It works by performing iterative computations on three related entities, i.e., reviewers, groups, and products. Eight Group Spam Behavior Indicators (GSBIs) are also proposed to capture suspicious group behaviors of spammers. Another competitor capable of working in unsupervised mode is the *learning to rank* approaches, in which the training rankings can be generated by sorting groups based on their h-CBMs/GSBIs in descending order. Two classic learning to rank algorithms - SVMRank [13] and RankBoost [14] - are included with default parameter settings. In summary, we have the following baselines for our comparison analysis:

GSRank (GSBI): As being tightly coupled with GSBIs, GSRank cannot work with other features such as h-CBMs.

SVMRank & **RankBoost**: Each of them is performed with GSBIs, h-CBMs, and both.

LCM: As LCM can work with other features, it is performed with GSBIs, h-CBMs, and both.

The output of each baseline is a score assigned to each reviewer in DL which represents his/her possibility of being colluders. The score of a reviewer is set to the maximum over the scores assigned to the groups (s)he belongs to (For LCM, the score assigned to each group is $p(\hat{z}|\hat{\phi}, \mathbf{\Phi}, \mathbf{Z})$ in Eq. (9)). The experiment is conducted using 5-fold cross-validation. As not being a trainable model, GSRank is evaluated by using one test folder each time. All the improvements of our method over the baselines are significant at $p < .01$ with paired t-test.

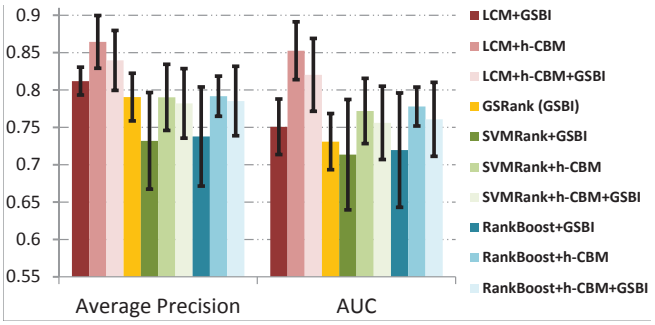
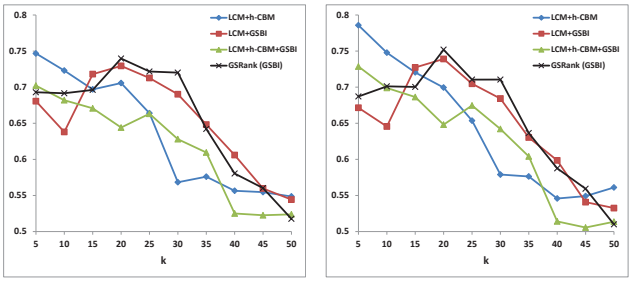


Figure 1: Cross validation comparison of LCM with other baselines. Error bars denote standard error of the mean.

The results are shown in Figure 1. It can be seen that LCM generally outperforms other methods in both metrics, with the best result achieved by using h-CBMs (LCM+h-CBM, AP=0.864, AUC=0.852). To compare different feature sets, we inspect the experiments using the same models. It shows that in all cases the performance can be improved by incorporating h-CBMs for training (by comparing X+GSBI with X+GSBI+h-CBM). The improvements range from 3.5% to 6.5% in AP and 5.8% to 9.3% in AUC. Also, the h-CBMs are shown to perform better than GSBIs in all cases (by comparing X+GSBI with X+h-CBM), by a margin of 6.4%-7.9% in AP and 8.1%-13.6% in AUC, which suggests the importance of characterizing colluders' homogeneous collective behaviors in practice. To compare the performance of different models, we inspect the experiments with the feature set fixed (X+GSBI). It can be seen that LCM shows superior performance over other baselines; the performance improves by 2.78% in AP and 2.59% in AUC compared to the second-place GSRank. Although the results of GSRank might be partially attributed to the evaluation paradigm where only one test folder is used as input (as GSRank is not trainable), the dominance of LCM over the other two trainable models (SVMRank and RankBoost) in both AP (by 10.9% and 10.1% resp.) and AUC (by 5.2% and 4.3% resp.) shows its competence in predicting collusive fraud.

C. Unsupervised Comparison on DU

As GSRank was experimented with DU [3], we then compare the collusion inference performance of LCM with GSRank on DU. Due to the lack of annotation, an unsupervised evaluation method [8] for ranking-based review fraud detection algorithms is used. Simply put, given a final ranking of reviewers with malicious ones ranked at the top and legitimate ones at the bottom, this approach first creates a pseudo annotated dataset by labeling the reviews posted by the top $k\%$ reviewers as positives and those posted by the bottom $k\%$ reviewers as negatives. Based on this corpus,



(a) F1-score (b) Accuracy

Figure 2: LCM vs. GSRank on DU. As the true number of colluders is unknown, we set k within [5, 50].

a standard *text classification* for fake reviews is conducted. The results will then be seen as an indirect evaluation of the original ranking algorithm. LCM can yield rankings by sorting reviewers based on the inference of their class labels, i.e., $q(z)$ in Eq. (8). Linear SVM is used as the text classification algorithm. 5-fold cross validation is performed for each model. The results (F1-score & accuracy) are shown in Figure 2.

We can observe similar trends in both metrics. In general, LCM achieves comparable results with GSRank (LCM+GSBI vs.GSRank), with the maximum differences from GSRank being 5.4% in F1-score and 5.6% in accuracy (k=10). This can be contributed by the effectiveness of GSBIs, which makes colluders well-separated from non-colluders in DU in terms of their behavior patterns [3]. LCM can also benefit from this separation; the larger it attains, the more accurately the collusion can be inferred by LCM. The utility of h-CBMs on DU can also be observed by inspecting the experiments involving the same model (LCM+X). It shows that when k is small ($<$15), h-CBMs perform the best. However, as k goes beyond 15, GSBIs begin to dominate. This shows that h-CBMs can effectively discover some portion of colluders in DU who have been missed by GSRank. As the superiority of h-CBMs over GSBIs does not last long (only with $k \in [5, 15]$), we speculate that DU may not be heavily attacked by collusive fraud because colludes in DU can still be treated as minorities or outliers and can effectively be caught by the anomaly-based features in GSBIs. Finally, the moderate performance of using GSBI+h-CBM suggests a compromise be made when combining h-CBMs with GSBIs.

V. Conclusions

In this paper, we identify the problem of detecting collusive fraud in online reviews from a stochastic perspective, and propose a novel statistical model called Latent Collusion Model (LCM) to model collusive review fraud. Not only can LCM perform collusion inference as unsupervised models, but it can also make collusion predictions as supervised models. Furthermore, multiple homogeneity-based collusive behavior measures (h-CBMs) are developed to capture the homophily inside colluders. The h-CBMs are complementary to existing collusion-oriented features in terms of handling stealthier collusive attacks. Experiments on two real-world review datasets show the effectiveness of h-CBMs and the superiority of LCM over state-of-the-art competitors in terms of collusion inference and predictive abilities. The source code and datasets used in this paper can be found at https://sites.google.com/site/homecxu/.

Acknowledgment

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] N. Jindal and B. Liu, "Opinion spam and analysis," in *WSDM*, 2008.

[2] M. Luca and G. Zervas, "Fake it till you make it: Reputation, competition, and yelp review fraud," *Harvard Business School NOM Unit Working Paper*, no. 14-006, 2013.

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

[4] C. Xu, J. Zhang, K. Chang, and C. Long, "Uncovering collusive spammers in chinese review websites," in *CIKM*, 2013.

[5] M. Rahman, B. Carbunar, J. Ballesteros, G. Burri, and D. H. P. Chau, "Turning the tide: Curbing deceptive yelp behaviors," in *SIAM SDM*, 2014.

[6] G. Wang, S. Xie, B. Liu, and P. S. Yu, "Identify online store review spammers via social review graph," *ACM TIST*, vol. 3, no. 4, p. 61, 2012.

[7] J. Ye and L. Akoglu, "Discovering opinion spammer groups by network footprints," in *ECML/PKDD*, 2015.

[8] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh, "Exploiting burstiness in reviews for review spammer detection," in *ICWSM*, 2013.

[9] A. Molavi Kakhki, C. Kliman-Silver, and A. Mislove, "Iolaus: Securing online content rating systems," in *WWW*, 2013.

[10] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 4, no. 4.

[11] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[12] C. I. Bliss, "The method of probits," *Science*, vol. 79, no. 2037, pp. 38–39, 1934.

[13] T. Joachims, "Optimizing search engines using clickthrough data," in *KDD*, 2002.

[14] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *The Journal of machine learning research*, vol. 4, pp. 933–969, 2003.