Bimodal Distribution and Co-Bursting in Review Spam Detection

Huayi Li, Geli Fei, Shuai Wang,
Bing Liu, Weixiang Shao
University of Illinois at Chicago
Illinois, USA
{lhymvp,garryfei,shuaiwanghk}@gmail.com
liub@cs.uic.edu
wshao4@uic.edu

Arjun Mukherjee[1], Jidong Shao[2]
[1]University of Houston
Texas, USA
arjun@cs.uh.edu
[2]Dianping Inc.
Shanghai, China
jidong.shao@dianping.com

ABSTRACT

Online reviews play a crucial role in helping consumers evaluate and compare products and services. This critical importance of reviews also incentivizes fraudsters (or spammers) to write fake or spam reviews to secretly promote or demote some target products and services. Existing approaches to detecting spam reviews and reviewers employed review contents, reviewer behaviors, star rating patterns, and reviewer-product networks for detection. In this research, we further discovered that reviewers' posting rates (number of reviews written in a period of time) also follow an interesting distribution pattern, which has not been reported before. That is, their posting rates are *bimodal*. Multiple spammers also tend to collectively and actively post reviews to the same set of products within a short time frame, which we call *co-bursting*. Furthermore, we found some other interesting patterns in individual reviewers' temporal dynamics and their co-bursting behaviors with other reviewers. Inspired by these findings, we first propose a two-mode Labeled Hidden Markov Model to model spamming using only individual reviewers' review posting times. We then extend it to the Coupled Hidden Markov Model to capture both reviewer posting behaviors and co-bursting signals. Our experiments show that the proposed model significantly outperforms state-of-the-art baselines in identifying individual spammers. Furthermore, we propose a co-bursting network based on co-bursting relations, which helps detect groups of spammers more effectively than existing approaches.

Keywords

Review Spam; Hidden Markov Model; Spam Groups

1. INTRODUCTION

Opinions in reviews are commonly used by individuals and organizations to make purchase decisions. Positive opinions often mean profits and fames for businesses and individuals, which unfortunately give strong incentives for fraudsters to secretly promote or to discredit some target products or services by writing fake/spam reviews. Such activities are called *opinion spamming* [16]. Several researchers have studied this problem [9, 19, 20, 26, 27]. Many review hosting companies such as Yelp and Dianping have also built their own review filtering systems to detect fake and low quality reviews from their product pages. These systems help alleviate the negative impact of fake reviews and greatly increase the cost of spamming. In order to hide their footprints and to be more effective, many spammers now work collectively to promote or to demote a set of target products [22, 35, 41]. Several researchers have worked on *collective* or *group spam detection* [31, 41, 42, 43]. Our work makes a significant advance due to two key findings from this research, *bimodal posting distribution* and *co-busting*, which we will detail shortly. They help us design better algorithms to detect both individual spammers and group spammers. We note that review spam is quite different from Web spam [6] or email spam [7], and much harder to spot even manually. See [16] for a detailed discussion and comparison. Review spam is also different in dynamics from Blog [17], network [15, 25], and tagging spam [18].

Although normal reviewers write reviews randomly, they have some tendency to write a few reviews after a period of inaction to summarize their recent experiences of using some services. Spammers have similar behaviors but for a different reason because they tend to participate in spam attacks/campaigns and write many reviews during a campaign but do not write much before or after that. Based on a large scale dataset (2,762,249 reviews and 633,381 reviewers) of Dianping's real-life filtered (fake or spam) reviews and unfiltered (genuine) reviews of all kinds of restaurants, we discovered that both spammers and non-spammers exhibit a *bimodal temporal posting distribution* in regard to their posting rates (number of reviews posted in a time period) but for different reasons as discussed above. Based on this finding, we propose a two-mode Labeled Hidden Markov Model (LHMM) to capture the bimodal behavior for detecting spammers (fake reviewers). The reviews of a reviewer in the order of their posting times form a chain. Hidden states of a reviewer at each posting time-stamp is either *active* or *inactive*. A reviewer in an active/inactive state means that he/she posts reviews in a fast/slow rate respectively.

Current research on collective or group spam detection is mainly based on the assumption that a set of spammers tend to write fake reviews together for the same set of products or services [31, 41, 42, 43], which we call *co-reviewing*. Reviews for a target also tend to form bursts due to fake review campaigns [8]. Co-reviewing may not necessarily mean *co-spamming* (i.e. working in collusion to spam the same set of products). Due to the advance of recommender systems, many consumers are likely to buy same products or to use same services. Through our analysis using the large Dian-

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052582

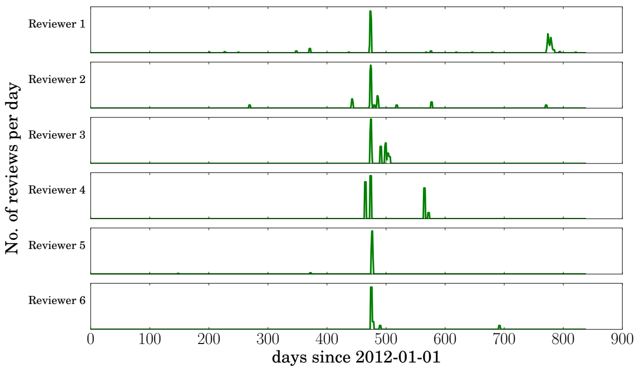


Figure 1: Examples of co-bursting behaviors

ping review dataset, we found that spammers tend to write reviews to the same restaurants not only "*collectively*" but also "*actively*" within a short period of time. Figure 1 gives an example of six spammers' daily numbers of reviews. At about day 480 (since 2012/01/01), all the six reviewers' were actively writing reviews. They were mostly inactive at other time periods. In addition to the temporal pattern, some reviews of the reviewers were written for the same set of restaurants. Such *co-bursting* patterns (several reviewers with bursts of reviews on the same set of targets) are prevalent in the Dianping dataset. We then extend the LHMM model to a Coupled Hidden Markov Model (CHMM) to detect spammers. CHMM has two parallel HMMs whose hidden states represent reviewer posting behaviors and co-bursting signals respectively.

Besides detecting individual spammers, the model's hidden states can be used to find spammer groups who work together in spam campaigns. We thus propose a co-bursting network of reviewers, which is used to identify collusion of reviewers. Clustering of reviewers in the network helps detect groups of spammers who work together. Reviewer clustering results indicate that our network is more effective in detecting spammer groups than the review-product network used in the existing work [31, 41, 42, 43].

We demonstrate the effectiveness of our methods by applying them to the large Dianping dataset. To our knowledge, this is the only large scale review spam dataset with spam and non-spam labels/classes and all reviews of each individual reviewer. Although there are yelp datasets with class labels [32, 35], they are much smaller and do not contain all reviews of each reviewer and are therefore not suitable for our spammer detection experiments. Our results show that the proposed models outperform state-of-the-art baselines in detecting both individual spammers and spammer groups.

In summary, this paper makes the following contributions:

1. To our knowledge, we are the first to discover the disparate bimodal posting rate distributions and state transition probability distributions of review spammers and non-spammers (detailed in Section 3.3). We propose a two-mode LHMM model to detect spammers by exploiting this bimodal distribution. Unlike HMM, the hidden states of LHMM are conditioned on the reviewer's class label, which allows the model to make predictions.
2. The paper further proposes the concept of co-bursting based on which the LHMM model is extended to the CHMM model by adding another parallel chain to exploit co-bursting signals. CHMM can then use both reviewer posting patterns and co-bursting behaviors of reviewers to produce a more powerful model.
3. The paper also proposes to use model hidden states to build a co-bursting network of reviewers for identifying groups of spammers who work together in spam campaigns. This results in a more effective method for detecting spammer groups than the current work based on co-reviewing [31, 41, 42, 43].

2. RELATED WORK

2.1 Bursty Reviews

Bursty reviews have been studied recently by several researchers. Fei et al. [8] studied review time-series for individual products. They assume spammers in a review burst of a product are working with other spammers. Similarly, Xie et al. [40] analyzed multiple review time-series of a single retailer including daily number of reviews, average rating, and ratio of singleton reviews. Their end task is to find the time intervals in which a spam attack happens to a retailer, which is quite different from our end task as we aim to find individual spammers and spammer groups. [37] explored temporal dynamics of spam in Yelp such as buffered and reduced spamming rates but does not model inter-arrival times. Other researchers applied various Bayesian approaches to detect anomalies in rating time-series [11, 12, 14, 44]. However, our model only requires the time stamp of each review and the byproduct of our model also allows us to detect spammer groups effectively as an extension to [23].

2.2 Classification and Ranking

Since our method is supervised based on available labels, we now review existing supervised learning methods for review spam detection. Review spam detection can be deemed as a binary classification or ranking problem. Ott et al. [33] built supervised learning models using unigrams and bigrams and Mukherjee et al. [32] added many behavioral features to improve it. [20] used semi-supervised learning. Others studied the task of psycholinguistic deception detection [30], computer-mediated deception in role-playing games [46] and so on. Besides, with only a small portion of labeled reviews, researchers pointed out that using Positive-Unlabeled Learning (or PU learning) [13, 21, 24, 36] outperforms traditional supervised learning. Since PU learning is not the focus of this work, we treat filtered reviews as positive and unfiltered reviews as negative. In the past few years, researchers also incorporated network relations into opinion spam detection. Most of them constructed a heterogeneous network of reviewers/reviews and products. Some of them employed HITS-like ranking algorithms [39], some applied Loopy Belief Propagation [1, 8, 35], and others utilized collective classification [21, 42]. In this work, we propose to build a network using co-bursting relations and it is shown to be more effective in capturing the spammers' correlations.

2.3 Spammer Group Detection

The second task of our paper is to identify collusive spammer groups. Although several methods have been proposed to uncover spam groups [4, 31, 41, 42, 43], they are all based on co-reviewing relations and have limitations in their assumptions. In section 4, we will compare our proposed approach based on the co-bursting network and the traditional co-reviewing network.

3. MODELING REVIEWERS' ACTIVITIES

3.1 Bimodal Distribution and Motivation

One of the reasonable models to capture the reviewer temporal activities is the Poisson Process which is a process where events occur continuously and independently at a constant average rate. However, after using the Poisson Process to model reviewers' posting behaviors, we found it quite inaccurate. We investigated the data by computing all the time intervals (denoted by Δ_i's) between adjacent reviews of spammers and non-spammers and plotted the histogram in Figure 2. Since the spam label in our data is on each review rather than each reviewer, we regard a reviewer as a spammer if at least 10% of his/her reviews are detected as fake/spam. We use 10% cutoff to allow for some errors in the data. We will discuss why Dianping's spam labels can be trusted in Section 5.1.

To our complete surprise, we discovered that the posting rate distribution is actually bimodal. Note that the x-axis of the figure is in log scale. More interestingly, this is true for both spammers and non-spammers. We can clearly observe two distinct peaks for spammers or non-spammers. As the Poisson distribution in this setting would typically model the spread of reviews in the next time step around a fixed average (i.e., there should be only one peak), this violates the bimodal distribution of inter-arrival time. Clearly, using a homogeneous Poisson Process is not suitable. To solve the problem, we follow the convention of [28] and propose a two-mode Labeled HMM to model Δ_i, which we discuss in the next section.

Further investigation showed that bimodal distribution is quite reasonable. First, non-spammers have the tendency to write a few reviews after a period of inaction to summarize their recent experiences after eating in some restaurants. Second, spammers participate in spam attacks/campaigns and write many reviews during a campaign but do not write much before or after that. We will make additional observations and discuss them in Section 3.3.

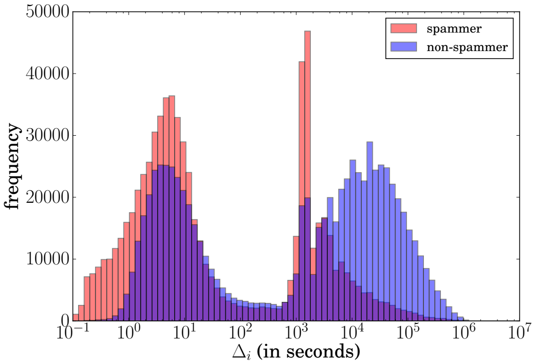


Figure 2: Bimodal distribution of time intervals between adjacent reviews. (Note: x-axis is in log scale)

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geqslant 0 \\ 0 & x < 0 \end{cases} \tag{1}$$

$$E(X) = \frac{1}{\lambda} \tag{2}$$

Let t_i, $i = 0, 1, \ldots, T$ be the time-stamps of a reviewer's reviews over a time period of interest. The inter-review duration or inter-arrival time between two adjacent reviews is denoted by Δ_i. By our assumption Δ_i is drawn from an exponential distribution with rate parameter λ.

$$\Delta_i \triangleq t_i - t_{i-1} \tag{3}$$

$$\Delta_i \sim Exp(\lambda) \tag{4}$$

3.2 User Level Behavior Modeling

Before we discuss our LHMM model in the next subsection, we first introduce how we model the temporal information with a two-mode HMM model. HMM is a model with a sequence of hidden states where one has only observed signals emitted from the hidden states. In our context, let t_i, $i = 0, 1, \ldots, T$ be the time-stamps of a reviewer's reviews over a time period of interest, and let the inter-review duration or inter-review arrival time $(t_i - t_{i-1})$ between two adjacent reviews be Δ_i. The hidden state Q_i represents *active* or *inactive* mode/state of reviewers and observed signals are the continuous variables Δ_i. Δ_i between time-stamp t_{i-1} and t_i may follow different exponential distributions depending on Q_i. Reviews in the active mode are written in a *fast* rate while reviews in the inactive mode are in a *slow* rate. Both rates are estimated from reviewers' review posting time and they correspond to the two modes/states. We now introduce the hidden states and properties of HMM.

Hidden States: We assume that a hidden state variable Q_i takes one of the two possible values $\{0, 1\}$ (two modes). $Q_i = 0$ denotes that the reviewer is in the inactive mode between time-stamp t_{i-1} and t_i while $Q_i = 1$ denotes that the reviewer is in the active mode. Our defined model is a first-order Markovian model which assumes Q_i depends only on Q_{i-1} and is independent of previous hidden states $Q_1, Q_2, \ldots, Q_{i-2}$. This approximation is proven reasonable in a great number of applications because it captures the short-term memory of human behaviors. Specifically, in our problem, we find strong correlations between consecutive time intervals Δ_{i-1} and Δ_i. Reviewers in active modes tend to be active and reviewers in inactive modes are more likely to stay inactive. The state transition probability matrix $\mathbb{A}$ is given in (5) where $a_{kj} = P(Q_i = j | Q_{i-1} = k)$, $k, j \in \{0, 1\}$. The initial state probability is a vector π and $\pi_j = P(Q_1 = j)$.

$$\mathbb{A} = \{a_{kj}\} = \begin{bmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \end{bmatrix} \tag{5}$$

Observation Density: Since the state variable is unobserved, we can only see the emitted time intervals between two consecutive reviews of a reviewer. In the two-mode HMM, Δ_i's can be either sampled from fast rate point process when $Q_i = 1$ or slow rate point process when $Q_i = 0$. The two different modes correspond to exponential distributions with rate parameters λ_0 and λ_1.

$$\Delta_i \sim \begin{cases} Exp(\lambda_0), & Q_i = 0 \\ Exp(\lambda_1), & Q_i = 1 \end{cases} \tag{6}$$

We now use (6) for drawing Δ_i with respect to Q_i, for $i \in [1, 2, \ldots, T]$. The emission probability distribution is denoted by $\mathbb{B} = \{b_j(\Delta)\}$ and $b_j(\Delta) = f(\Delta; \lambda_j) = \lambda_j e^{-\lambda_j \Delta}$ is the probability of observing some Δ at state j, where $j \in \{0, 1\}$ and λ_j is the rate parameter of Poisson distribution. Now we can formulate the joint probability of the observations $\Delta_{1:T}$ and hidden states $Q_{1:T}$:

$$\begin{aligned} & P(Q_{1:T}, \Delta_{1:T}) \\ &= P(Q_1, Q_2, \Delta_2, \ldots, Q_T, \Delta_T) \\ &= P(Q_1) \prod_{i=2}^{T} P(\Delta_i | Q_i) \prod_{i=2}^{T} P(Q_i | Q_{i-1}) \end{aligned} \tag{7}$$

One of the three basic problems of HMM is called the decoding problem which aims to estimate the most likely state sequence in the model given the observations (8). Identifying the hidden states helps to better understand spammers and their collusive behaviors.

$$
\begin{aligned}
Q^*_{1:T} &= \underset{Q_{1:T}}{\operatorname{argmax}}\, P(Q_{1:T}|\Delta_{1:T}) \\
&= \underset{Q_{1:T}}{\operatorname{argmax}}\, P(Q_{1:T}, \Delta_{1:T})
\end{aligned}
\tag{8}
$$

A naive approach to examine all possible state assignments has a running time $O(T \cdot 2^T)$ because there are totally 2^T possibles combinations and for each such combination, it requires $O(T)$ time to calculate the product of probabilities. Fortunately, we can employ an efficient dynamic programming algorithm named Viterbi [10] to reduce the time complexity to $O(T \cdot 2^2)$ or simply $O(T)$. Let's define a vector

$$
\delta_i(j) = \max_{Q_{1:i-1}} P(Q_{1:i-1}, Q_i = j, \Delta_{1:T}) \tag{9}
$$

for storing the maximum joint probability along a single path from Q_1 to Q_{i-1} when the current assignment is $Q_i = j$. On initialization, we set $\delta_1(j) = \pi_j b_j(\Delta_1)$ for $j \in \{0, 1\}$. Then we iteratively calculate $\delta_i(j)$ using (10) and finally the last state Q^*_T of the most likely state sequence is the one that maximizes (11). Starting from the last state, the sequence of most likely state sequences can be back-tracked through (12).

$$
\delta_i(j) = b_j(\Delta_i) \max_{k \in \{0,1\}} \big(\delta_{i-1}(k) a_{kj}\big),\ 2 \leqslant i \leqslant T,\ j \in \{0, 1\} \tag{10}
$$

$$
Q^*_T = \underset{j \in \{0,1\}}{\operatorname{argmax}}\, \delta_T(j) \tag{11}
$$

$$
Q^*_{i-1} = \underset{j \in \{0,1\}}{\operatorname{argmax}}\, \delta_i(j)\, a_{jQ^*_i},\ \ 2 \leqslant i \leqslant T \tag{12}
$$

Identifying the state sequence for each reviewer is useful in the sense that reviews from active and inactive states have different impact on calibrating spammers' behaviors. We will show in section 3.4 that spammers tend to collaborate in active states.

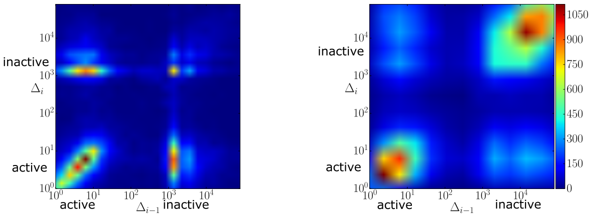
3.3 Labeled Hidden Markov Model

Note that the two-mode HMM is mainly for capturing the temporal dynamics and thus unsupervised. Now we incorporate the label information to measure and classify spammers and non-spammers. Recall that we plotted the histogram of all the time intervals of adjacent reviews for spammers (red) and non-spammers (blue) in Figure 2, which induce the following important observations:

- Bimodal distribution for both classes: The reviews of both spammers and non-spammers show a bimodal distribution. The centers of the two peaks are far apart from each other indicating distinct two-mode states of review writing patterns. Note that we use the log scale for the x-axis. For reviews of spammers, active states may be the result of aggressive spam activities from a group of spammers in collusion. For reviews of non-spammers, active states are likely to happen when normal reviewers write a few reviews after a period of inaction to summarize their recent experiences afterwards.

- Distinct distributions for active and inactive modes: Since the x-axis of the plot is in log-scale, we can see the histogram for reviews of non-spammer have much longer tails than those of spammers. This means that a lot of reviews of non-spammers are written in inactive mode. Besides, there are many more reviews from spammers in active mode especially less than 100 seconds.

- Disparity of mean of time intervals: For both classes, we simply run the k-means algorithm on the time intervals (log-scale) and compute the mean of inter-arrival times. We found that for both active and inactive states, the mean of the time intervals of non-spammers' reviews are about two to three times longer than that of spammers' reviews showing a rather normal reviewing activity as the latter are tend to be bursty [8].



(a) spammers (b) non-spammers

Figure 3: Heatmap of consecutive time interval pairs (in seconds). Each point corresponds to (Δ_{i-1}, Δ_i) for some reviewer.

In addition to the disparity of the emission probability, we also find different transition patterns between two states for spammers and non-spammers. For each of the two reviewer classes, we computed the consecutive time intervals between reviewer's reviews and visualized the distribution of all pairs of previous time interval Δ_{i-1} and current time interval Δ_i in the heatmap in Figure 3. In both sub-figures, we can easily see four regions that correspond to four types of state transitions. The lower left region means the transition that the active state at t_{i-1} remains active at t_i and likewise, the upper right corner are those states remaining inactive. The upper left region corresponds to inactive states changed from active states while the lower right one is the opposite. We can make the following interesting observations:

- In the lower left corner of Figure 3(a), there is a strong positive correlation between Δ_{i-1} and Δ_i for spammers when states remain active whereas the correlation between non-spammers in Figure 3(b) is very weak. This may be because even though different spammers exhibit different posting rates while in the active state, the posting rates for a single spammer will not change much. But posting rates of an ordinal reviewer in the active state at different timestamps may vary. As a consequence, we can see the hot area in the lower left region forming a line along the diagonal for Figure 3(a) but not for Figure 3(b).

- In Figure 3(a), when spammers' states change from inactive to active (lower right region), their active states are different from each other. This is due to the fact that when spammers are activated, they begin to post fake reviews in various rates because spammers from different campaigns may behave differently. However, when spammers' states transit from active to inactive (upper left region), their inactive states are very similar to each other because once a spam campaign is over, the time intervals between a

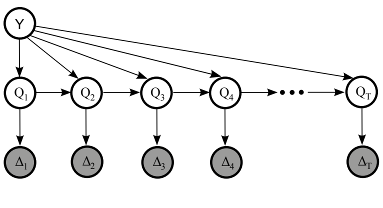


Figure 4: Representation of Labeled Hidden Markov Model

spammer's last active review and the next inactive review follow similar patterns.

- On the contrary, in Figure 3(b), ordinary reviewers who transit from inactive states to active states write reviews actively in similar rates to each other (lower right region), because they are not driven by any campaigns or motivations. The variance of the time interval between their last inactive review and next active review is small. And because normal reviewers "hibernate" differently due to their own habits. The time it takes for each of them to write a review after writing the last active review is significantly different. Thus their transition from active state to inactive state (upper left region) takes different amount of time.

- In the upper right corner of Figure 3(a) and Figure(b), we find that spammers "rest" in a similar rate concentrating in a small region. However, when normal reviewers are resting, the time intervals between reviews spread out the entire upper right corner. Clearly patterns for non-spammers are more natural and organic.

Based on the discovery of the major differences between emission probability and transition probability of HMM that ran on two classes of reviewers, we propose a novel extension to the two-mode HMM and call it the Labeled Hidden Markov Model (LHMM) which incorporates the class labels available in our dataset. The parameters of LHMM are learned from the training data which is then used in prediction on the testing data using the Baum-Welch method [34]. Based on the original two-mode HMM model, we introduce a new binary variable Y to represent the classes or labels as shown in Figure 4. $Y = +$ stands for spammers and $-$ for non-spammers. The variable Y plays a significant role in the generating process of HMM. The transition probability matrix $\mathbb{A}$ is extended to $\mathbb{A}^+$ and $\mathbb{A}^-$ for spammers and non-spammers respectively. The set of rate parameters $< \lambda_0, \lambda_1 >$ now becomes $< \lambda_0^+, \lambda_0^-, \lambda_1^+, \lambda_1^- >$. Consequently, the emission probability is dependent on the reviewer class Y (13). All the paremeters are learned from our data with labels from Dianping.

$$\Delta_i \sim \begin{cases} Exp(\lambda_0^Y), & Q_i = 0 \\ Exp(\lambda_1^Y), & Q_i = 1 \end{cases} \tag{13}$$

In order to predict the value of Y given the observations $\Delta_{1:T}$, we need to use Bayesian theorem. The most probable value that the class variable takes is the one that better explains or generates the observations. Thus we have the following:

$$\begin{aligned} y^* &= \underset{y}{\operatorname{argmax}}\, P(Y = y|\Delta_{1:T}) \\ &= \underset{y}{\operatorname{argmax}}\, \frac{P(\Delta_{1:T}|Y = y) \cdot P(Y = y)}{P(\Delta_{1:T})} \end{aligned} \tag{14}$$

The denominator $P(\Delta_{1:T})$ in (14) is a constant term regardless of y, so we can simply drop it. The prior probability of the class variable $P(Y)$ can be easily computed by counting. The difficult part is the conditional probability $P(\Delta_{1:T}|Y)$. Recall that equation (7) is the joint probability of observations and hidden states, the conditional probability can be calculated by marginalizing the hidden states:

$$\begin{aligned} & P(\Delta_{1:T}|Y) \\ &= \sum_{Q_{1:T}} P(Q_{1:T}, \Delta_{1:T}|Y) \\ &= \sum_{Q_{1:T}} P(Q_1|Y) \prod_{i=2}^{T} P(\Delta_i|Q_i, Y) \prod_{i=2}^{T} P(Q_i|Q_{i-1}, Y) \end{aligned} \tag{15}$$

By its direct definition, the time complexity is $O(T \cdot 2^T)$. Fortunately, another dynamic programming algorithm named Forward-backward method [3, 34] can largely reduce it to linear time. Similar to Viterbi, the Forward-backward method caches intermediate results to facilitate the computation.

We define a variable $\alpha_i(j|y) = P(\Delta_{1:i}, Q_i = j|Y)$ to store the joint probability of observations and $Q_i = j$ with all previous states $Q_{1:i-1}$ marginalized given Y. To do so, we first initialize $\alpha_1(j|y) = \pi_j^y \cdot b_j(\Delta_1|y)$, $j \in \{0, 1\}$ and then iteratively solve $\alpha_i(j|y)$, for $i = 2, \ldots, T$.

$$\alpha_i(j|y) = b_j(\Delta_i|y) \sum_{k \in \{0,1\}} \alpha_{i-1}(k)\, a_{kj}. \tag{16}$$

After that, we can get $P(\Delta_{1:T}|Y) = \sum_j \alpha_T(j|y)$ easily.

3.4 Coupled Hidden Markov Model with Co-bursting Behaviors

Recall in Figure 1, we found that when a restaurant has bursty reviews arriving at some point, many spammers are likely to be actively writing reviews to it as well as to many other restaurants. We call it co-bursting (i.e., a group of reviewers who have bursty reviews, some of which are posted to the same set of restaurants in a short period of time) as opposed to co-reviewing (reviewers reviewing the same set of restaurants together).

With respect to a specific review at time t to a restaurant S from a certain reviewer, we consider 6 intuitive co-bursting metrics to quantify co-spamming activities from other reviewers who happen to write reviews to the same business within a time window $< t - \omega,\ t + \omega >$.

1. **No. of co-reviews**: This metric simply counts the number of reviews of other reviewers' to the same restaurant within the time window.

2. **No. of spam co-reviews**: After running the LHMM model, we classify each review into spam or non-spam. This metric is similar to the first one except that only spam reviews are counted.

3. **No. of co-reviews when restaurant is active**: The metric is also similar to the first one except that it is conditioned on whether the restaurant of interest has bursty reviews.

4. **No. of spam co-reviews when restaurant is active**: Similarly to the third metric, but only spam reviews are included.

5. **No. of co-reviews when reviewer is active**: Similar to the first metric, this one only counts co-reviews when their reviewers are in the active state.

6. **No. of spam co-reviews when reviewer is active**: This metric considers only spam co-reviews from active reviewers.

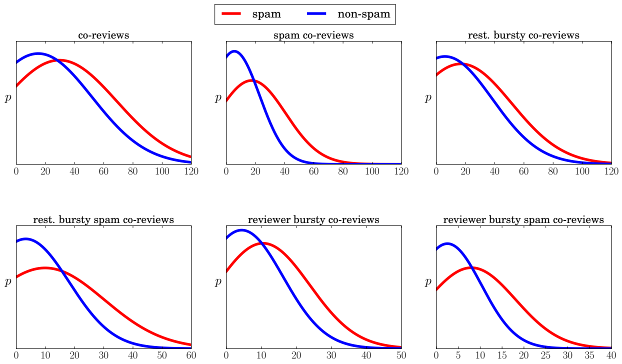


Figure 5: PDF of Gaussian distribution of co-bursting features

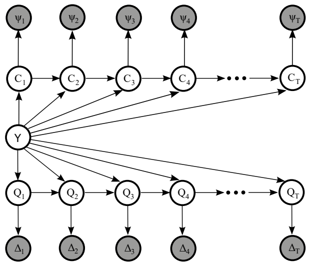


Figure 6: Representation of Coupled Hidden Markov Model

Similarly to the single chain HMM, the above-mentioned metrics are observations of co-spamming activities which can be considered to be generated through two different modes (co-bursting mode or normal mode). We assume each of the 6 metrics of a review is generated from a Multivariate Gaussian distribution of two set of parameters corresponding to the two different modes. Plots in Figure 5 demonstrate the great disparity of co-bursting metrics between spam and non-spam reviews. In general, spam reviews are associated with more intensive co-bursting activities for all six dimensions than non-spam ones. Inspired by this discovery, we propose to extend the LHMM model to incorporate co-bursting relations to better model reviewers' collective behaviors. In Figure 6, we add another chain to represent the sequence of observed co-bursting metrics of a reviewer. Observed co-bursting signals at t are denoted as Ψ_t which is generated from the underlying Gaussion distribution at mode C_t where $C_t \in \{0, 1\}$. $C_t = 1$ means the co-bursting mode. We call this model Coupled Hidden Markov Model (CHMM) as it contains two parallel HMM chains corresponding to each other. With the extra knowledge from co-bursting, the estimation of reviewer's class is more accurate which we will show in the experiment section. Under such a framework, the inference problem becomes finding the best reviewer label Y that maximizes the joint probability with observed intervals and co-bursting signals $P(\Delta_{1:T}, \Psi_{1:T}, Y)$. Again, we can solve the inference problem as below by eliminating hidden variables $Q_{1:T}$ and $C_{1:T}$ using forward propagation.

$$\begin{aligned}
y^* &= \underset{y}{\operatorname{argmax}}\, P(Y = y|\Delta_{1:T}, \Psi_{1:T}) = \underset{y}{\operatorname{argmax}}\, P(\Delta_{1:T}, \Psi_{1:T}, Y = y) \\
&= \underset{y}{\operatorname{argmax}}\, P(\Delta_{1:T}|y) \cdot P(\Psi_{1:T}|y) \cdot P(y) \\
&= \underset{y}{\operatorname{argmax}} \sum_{Q_{1:T}} P(Q_{1:T}, \Delta_{1:T}|y) \sum_{C_{1:T}} P(C_{1:T}, \Psi_{1:T}|y) \cdot P(y) \\
&= \sum_{Q_{1:T}} P(Q_1|Y) \prod_{i=2}^{T} P(\Delta_i|Q_i, Y) P(Q_i|Q_{i-1}, Y) \\
&\quad \cdot \sum_{C_{1:T}} P(C_1|Y) \prod_{i=2}^{T} P(\Psi_i|C_i, Y) P(C_i|C_{i-1}, Y)
\end{aligned} \tag{17}$$

4. DETECTING SPAMMER GROUPS

Since we discovered co-bursting is prevalent in spammers, it is natural to consider using it to detect collusion of spammers or spammer groups. In this section, we discuss how hidden states estimated from any of our models can be used to detect such groups by creating a co-bursting network. Since the traditional co-reviewing network has no knowledge of reviewers' label, for the fairness of comparison, we only apply two-mode HMM to construct the co-bursting network. Group spamming refers to a group of reviewers writing fake reviews together to promote or to demote some target products. A spam group or a spam community is more damaging than a single individual spammer as members of a group can launch a spam attack together in a stealth mode, and due to multiple members, a group can take total control of the sentiment on a product. Each individual spammer may not look suspicious in this case, but a bigger picture of all of them sheds light on the collusive behaviors of a spam community. Thus, identifying such groups is important.

Previous studies on spammer groups in [31, 42] proposed to use Frequent Itemset Mining (FIM). They treat reviewers as items and the businesses/products as transactions. Their idea is to extract groups of reviewers who have reviewed multiple products together. But it suffers from a few drawbacks.

- Computationally expensive: The problem is equivalent to finding all complete bi-partite subgraphs in the reviewer-product network, which is NP-hard. Using a high support threshold in FIM will find only a few extreme cases (low recall), while low support causes combinatorial explosion especially in large datasets where there are millions of reviewers and thousands or more of products.
- Failure to capture loosely connected subgraphs: Itemsets in FIM correspond to a complete subgraph. But it is not necessarily true that every spammer should connect to all the products reviewed by other members in the same group.
- Co-reviewing doesn't mean co-spamming: There is a good chance that genuine reviewers may happen to co-review some popular products/businesses. Nowadays recommendation systems are also suggesting consumers to buy similar products. The assumption that co-reviewing leads to co-spamming is too strong.

Since our Hidden Markov Model gives a good estimation of hidden states for all the reviews, we propose to construct a co-bursting network based on the active state of reviews, as co-bursting relations are good indicators of group spamming. Intuitively, the co-bursting network is more representative of the collective spamming behaviors and is thus more effective at capturing relationships between spammers than the review-product network, which were used to detect spammer groups previously in [31, 42]. Because it is much cleaner than reviewer-product network, the chance of random correlations is much lower. Thus it is useful to measure the degree of collaboration between spammers.

We denote the co-bursting network as $F = \{F_{uv}\}^{n \times n}$, where n is the total number of reviewers (nodes). The weight of the undirected edge of node u and v is F_{uv} representing the number of times reviewer u and reviewer v co-burst within a time window ω to some restaurant (rest). In our setting, we choose $\omega = 3$ days. r_i.state means the hidden state of review i and r_i.t is the time when it is posted.

$$\begin{aligned}
F_{uv} = \Big| (r_i, r_j) : r_i \in R^u, r_j \in R^v,\ r_i.\text{rest} = r_j.\text{rest}, \\
|r_i.\text{t} - r_j.\text{t}| < \omega,\ \ r_i.\text{state} = r_j.\text{state} = 1 \Big|
\end{aligned} \tag{18}$$

A straightforward approach to construct the co-bursting network using equation 18 is very inefficient. Thus, in Algorithm 1 we propose to use a B+ tree and a hashtable to facilitate the computation. We first group reviews by its reviewer and run our proposed two-mode HMM model to get estimated states for all reviews (Line 1-3) and then we build a B+ tree for each restaurant to support range queries on the timestamps (Line 4-6). We maintain a hashtable to store the number of times a pair of reviewers co-burst which is calculated efficiently from Line 7-15. The overall run-time for the last querying step is $O(m \times log(p))$ where m is the total number of re-

Algorithm 1: Construct the co-bursting network efficiently

```
Input: a set of reviews $R$, a set of reviewers $U$, a set of
       restaurants $S$, time window $\omega$
Output: the co-bursting matrix $F$.
for each $u \in U$ do
    $R^u = |r \in R : r.\mathrm{reviewer} = u|$
    Run two-mode HMM on $R^u$ to get the estimated state of
      each of his reviews stored as $r.\mathrm{state}$
for each $s \in S$ do
    $R^s = |r \in R : r.\mathrm{restaurant} = s|$
    Build a B+ tree $T^s$ for $R^s$ indexing on the posting time of
      reviews.
Create a hashtable $H$ to store the number of times of
  co-bursting for a pair of reviewers
for each $u \in U$ do
    for each $r \in R^u$ do
        $s = r.\mathrm{restaurant}$
        query B+ Tree $T^s$ to get reviews for restaurant $s$
          posted between $< r.t - \omega,\ r.t + \omega >$ which are
          denoted as $C$.
        for each review $c \in C$ do
            if $r.state$ = $c.state$ = $1$ then
                $i = r.\mathrm{reviewer}$,   $j = c.\mathrm{reviewer}$
                $H_{i,j} = H_{i,j} + 1$
Convert $H$ to sparse matrix $F$ and output $F$
```

views in the dataset and p is the average number of reviews written to a restaurant. Because $log(p)$ is a small constant, our proposed algorithm is linear to the number of reviews and it is scalable to large datasets for commercial review websites. Once the co-bursting network is constructed, graph clustering can be used to find clusters, which are spammer groups (see the next section).

5. EXPERIMENTS

5.1 Fake Review Datasets

Jindal and Liu [16] released the first opinion spam dataset crawled from Amazon. They treated duplicate and near-duplicate reviews as fake/spam. However, it misses many fake reviews that are not duplicated. Ott et al. [33] used Amazon Mechanical Turk (AMT) to crowdsource fake hotel reviews. Their dataset contains only 1,600 reviews which is small and does not have reviewer's posting time and other information. Other researchers [32, 35] reported analyses of the Yelp filter based on reviews they crawled. They assumed those reviews which are filtered by Yelp are spam and compiled two datasets respectively: Yelp-Chicago [32] and YelpZip [35]. However, these datasets do not have all reviews of each reviewer as they crawled Yelp reviews based on products. On average, each reviewer has only 1.9 reviews in YelpChicago and 2.9 reviews in YelpZip. They are thus not suitable for our work because we need all reviews of a reviewer with review posting times.

Our dataset from Dianping consists of reviews of popular restaurants in Shanghai, China from Nov. 2011 to Apr. 2014. It includes all reviews of each reviewer. Since we model reviewers' behaviors, for reliability we only consider reviewers with at least 10 reviews. Under this criterion, the dataset still contains 1,582,069 reviews from 67,698 reviewers. Each review is labeled as spam or non-spam using Dianping's commercial spam filter. We regard a reviewer as a spammer if s/he has at least 10% of his/her reviews detected as fake/spam by Dianping. This cutoff allows for some errors in Dianping's detection. Also, among the reviewers with at least one spam review, only 2.3% of them have less than 10% spam reviews.

Dianping's review spam labels can be trusted because of the following reasons: Dianping has a feedback system allowing reviewers to complain. If they complain that their "genuine" reviews are removed, Dianping will send them the evidences for removing their reviews. Dianping's record shows that complaints are rare. Dianping also has an expert team that manually evaluate sampled reviews constantly. Dianping's CTO claimed that they have used over 100 algorithms and the accuracy of their system is about 95%[1]. Therefore, only the Dianping dataset is suitable for our experiments which require a complete history of reviewers' activities.

5.2 Spammer Classification

In our experiment, reviews are grouped by reviewers and sorted in the order when they are posted. The parameters of our models are learned from training data which are then used for prediction on the testing data to detect spammers or fake reviewers. We first compare LHMM and CHMM with existing supervised learning methods. Although there are many recent progresses on review spam, due to lack of ground truth, most of the studies are semi-supervised or unsupervised grounded on the authors' intuitions [8, 39, 40, 43]. Since our approach is supervised, it is fair to compare with supervised learning models as listed below.

1. **SVM(ngram)** [33]: Ott et al. built a Support Vector Machines classifier using text features including unigrams and bigrams.
2. **SVM(BF)** [32]: Mukherjee et al. proposed many behavioral features including the number of reviews per day, rating deviation, content similarity, etc. They showed that only using reviewers' behavior features (BF) achieves better performances.
3. **SVM(ngram+BF)** [32]: Mukherjee et al. combined behavioral features with ngram text features to improve the results.
4. **PU-LEA** [13]: The first Positive-Unlabeled learning model applied in review spam detection is PU-LEA. PU learning usually outperforms traditional supervised learning when there are hidden positive instances inside the negative data. This is the case because there should be spam reviews that are not discovered by Dianping.
5. **LHMM (UT)**: Here we want to show how important the transition probability of the single chain Labeled HMM (LHMM) is, so we use the uniform transition (UT) probability in LHMM rather than that learned from data.
6. **LHMM**: The proposed LHMM model whose observed variables are time intervals (Figure 4). Transition probabilities are learned from the training data using the Baum-Welch method.
7. **LHMM (MG)**: Just as LHMM, but the observed variables are co-bursting signals from the multivariate Gaussian distribution. We also evaluate this variant to see how LHMM using co-bursting signals alone performs.
8. **CHMM**: This is the Coupled HMM model proposed in Figure 6 with two parallel HMMs that incorporate both the reviewer's posting behavior and co-bursting behaviors from other reviewers.

The effectiveness of all models are evaluated using the standard Accuracy, Precision, Recall and F1-score based on five-fold cross validation. We can observe that all LHMM based models markedly outperforms the baselines in review spammers detection as shown in Figure 7. It is worth noting that the largest gain of our model is recall. Because some spam accounts may exhibit mixed behaviors which confuse those classifiers based on language and behavior features, whereas our proposed LHMM can successfully model

[1] http://weibo.com/2235685314/BaoyXqlgt?type=comment

such temporal dynamics. Compared with LHMM(UT) which uses uniform transition probability, LHMM can achieve better results as it learns the transition probability from the data, which well captures the transitional behaviors as shown in Figure 3. Other than the final CHMM model, LHMM has the highest recall and LHMM(MG) achieves the best precision. Since they are modeling reviewers' behaviors from different angles, the CHMM model which is a joint model of LHMM and LHMM(MG) has the best overall F1 score. These results indicate the strong impact of reviewers' posting dynamics and co-bursting signals.

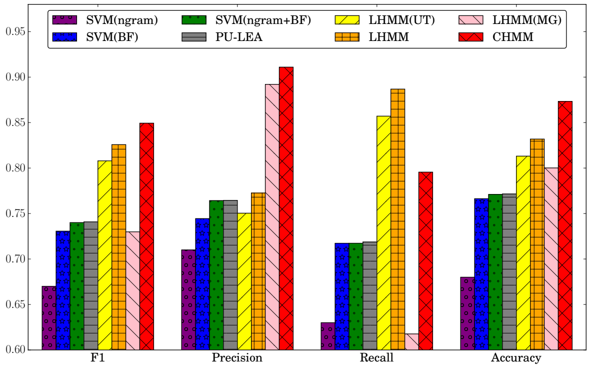


Figure 7: Model performance in Accuracy(A), Precision(P), Recall(R) and F1-score(F) (Positive class is spammer).

According to Dianping, using raised accounts to spam (write fake reviews) is quite popular in their data. Raised accounts are those accounts that review normally for a period of time to accumulate credits or reputation. They are then used to write fake reviews to avoid detection by simple algorithms. For such raised high reputation accounts, businesses usually have to pay four times more to get fake reviews in the underground market[2]. In the Dianping dataset, over 40% of the spammers fall into this category. Figure 8 exemplifies the daily reviews counts of three raised accounts detected by our model. Clearly, there are two distinct phases: one is the *farming phase* when the account behaves normally and randomly posts reviews to accumulate credits; the other phase is the *harvest phase* when the raised account aggressively posts spam reviews. We further investigated the effectiveness of our model in detecting raised accounts. Our proposed method successfully detected 85.41% of all the raised accounts in the data.

5.3 Spammer Group Clustering

The ground truth of spammers' group affiliation is very hard, if not impossible, to obtain. We resort to evaluate the clustering quality instead. This is reasonable because the co-bursting network already reflects strong correlations between reviewers. It is very likely that reviewers in high quality clusters belong to true spammer groups. We then apply some existing clustering algorithms to cluster the network, and evaluate the results [29, 45] to see whether the clusters catch spammers based on spammer labels in our data.

Since our goal is to validate that the co-bursting network is more intuitive and helpful in quantifying reviewer collaborations than co-reviewing which is more noisy, we build two types of networks: co-review network using reviewer-product relations [31, 39, 42] and co-bursting network using reviewers' hidden states according to our definition in equation (18). Then we apply three efficient clustering algorithms that are suitable for the scale of our dataset (using all the data): Louvain Method [5], Kmeans and a hierarchical clustering algorithm from recent work [43]. We employ open-source libraries Networkx[3] and scikit-learn[4] to implement those methods and they all support finding the optimal number of clusters.

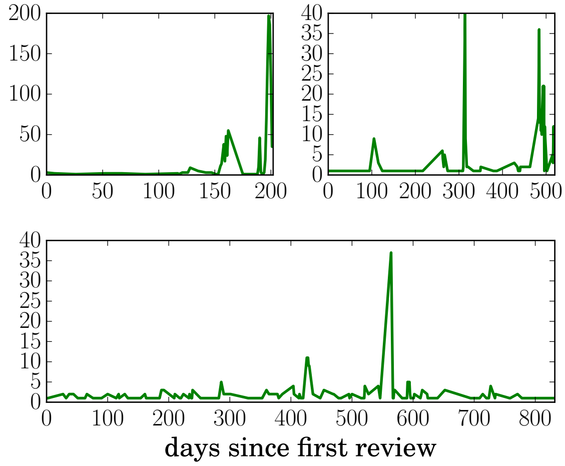


Figure 8: Number of daily reviews of three raised accounts

Table 1: Evaluation of models' performances

Method	Purity		Entropy	
	co-review	co-burst	co-review	co-burst
Louvain	0.69	**0.83**	0.87	**0.67**
Kmeans	0.72	**0.86**	0.81	**0.73**
Hierarchical	0.72	**0.88**	0.82	**0.76**

Numbers in bold indicate better performance

We use two important metrics to evaluate the clustering results: *Purity* and *Entropy*, which are widely used measures of cluster quality based on ground truth labels [2]. Purity [29] is a metric in which each cluster is assigned to the class with the majority vote in it and the accuracy of this assignment is the number of correctly assigned instances divided by the total number of instances N.

$$purity(C, Y) = \frac{1}{N} \sum_k max_j |y_j \cap c_k| \qquad (19)$$

where $C = \{c_1, \ldots, c_k\}$ is the set of cluster ids and $Y = \{y_1, \ldots, y_j\}$ is the set of reviewers' real class labels. c_k is interpreted as the set of reviewers in cluster k and y_j is the set of reviewers whose label is j. The higher purity score means a purer cluster. Entropy [38] measures the uniformity of a cluster. The entropy of all clusters is the weighted sum of entropy of each cluster:

$$entropy = -\sum_k \frac{n_k}{N} \sum_j P(j,k) \log_2 P(j,k) \qquad (20)$$

where $P(j,k)$ is the probability of finding a reviewer of class j in cluster k. The quality of a cluster improves as the entropy de-

[2] http://finance.sina.com.cn/consume/puguangtai/20120904/061913036552.shtml

[3] http://perso.crans.org/aynaud/communities/

[4] https://goo.gl/olFLKD

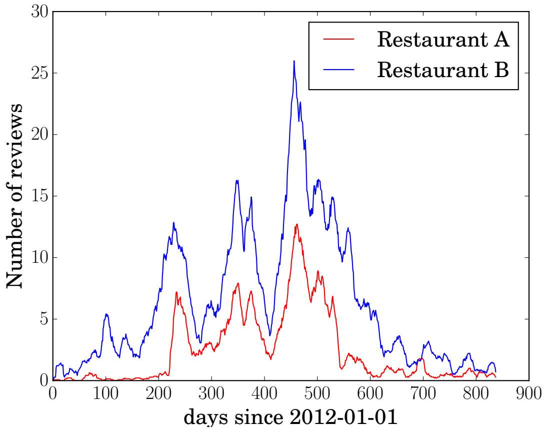


Figure 9: Strong positive correlation between daily fake reviews of two restaurants that are only 100 meters apart

creases. In Table 1, we list the purity and entropy of the clustering results. For each clustering algorithm, clusters computed from the co-bursting graph are markedly better than those from the co-reviewing graph. Such finding confirms our intuition.

5.4 Case Study: Restaurants Co-bursting

The collective spamming behaviors from spammers result in a similar view from the perspective of restaurants. Since there are many spammers actively writing reviews to a set of restaurants to promote some businesses, it is very likely to see those restaurants' time-series of daily (fake) reviews to co-burst as well. Figure 9 shows an example, which has two restaurants that are only within 100 meters apart. We found that there is a very strong positive correlation between their numbers of daily reviews (applied with 14-day moving average) and we noticed that especially in the bursty regions, their correlation is the highest which indicates the co-bursting behaviors of the restaurants. We further investigated whether they are indeed promoted by some spammer community or at least whether they were promoted by the same set of common spammers. There are overall 3196 reviewers for restaurant A and 8686 reviewers for restaurant B and interestingly they share 1166 reviewers. From April, 2013 to May, 2013 which correspond to the highest spike of the two time-series, we found 311 reviewers wrote fake reviews to restaurant A and 591 reviewers to restaurant B and among those reviewers 139 reviewers wrote fake reviews to both restaurants. Spammer groups often proactively look for business owners to convince them to use their services. It is not surprising to see that they can help both restaurants who are competitors in the same business zone because it is easy to convince a business owner if his rival is already working with them. This explains the high correlation between their bursty regions. In summary, such views from the perspective of restaurants' bursts provide a different angle to show the intense collusion among spammer communities and explains the important reason why our model can detect hard case scenarios where traditional linguistic and behavioral features may not work well.

6. CONCLUSION

In this work, we first conducted a series of analyses using Dianping's real-life dataset with spam labels. The analyses showed bimodal distributions of review posting rates and some major differences of temporal patterns of spammers and non-spammers. Beyond that, there also exists clear distinction in their state transitions. Based on the discoveries, we proposed a two-mode Labeled HMM to model reviewers' posting activities for detecting review spammers. The parameters are learned from data and hidden states of reviews are inferred from our model. In addition, we found many spammers happen to actively write fake reviews to the same restaurants together in a short period of time, so we defined a set of co-bursting metrics and extended our model to a Coupled HMM model. Hidden states estimated from our model are also good clues for discovering collusive spammers whose collective behaviors are well captured by co-bursting. Our experimental results showed superior results compared to the state-of-the-art baselines.

7. ACKNOWLEDGMENT

This work was supported in part by grants from National Science Foundation (NSF) under grant no. IIS-1407927 and IIS-1650900. Arjun Mukherjee was supported in part by NSF CNS-1527364.

References

[1] L. Akoglu, R. Chandy, and C. Faloutsos. Opinion fraud detection in online reviews by network effects. In *ICWSM*, pages 2–11, 2013.

[2] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486, 2009.

[3] L. E. Baum, J. A. Eagon, et al. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bull. Amer. Math. Soc*, 73(3):360–363, 1967.

[4] A. Beutel, W. Xu, V. Guruswami, C. Palow, and C. Faloutsos. Copycatch: Stopping group attacks by spotting lockstep behavior in social networks. In *WWW*, pages 119–130, 2013.

[5] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[6] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: Web spam detection using the web topology. In *SIGIR*, pages 423–430, 2007.

[7] P.-A. Chirita, J. Diederich, and W. Nejdl. Mailrank: using ranking for spam detection. In *CIKM*, pages 373–380, 2005.

[8] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh. Exploiting burstiness in reviews for review spammer detection. In *ICWSM*, pages 175–184, 2013.

[9] S. Feng, L. Xing, A. Gogar, and Y. Choi. Distributional footprints of deceptive product reviews. In *ICWSM*, pages 98–105, 2012.

[10] G. D. Forney Jr. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[11] N. Günnemann, S. Günnemann, and C. Faloutsos. Robust multivariate autoregression for anomaly detection in dynamic product ratings. In *WWW*, pages 361–372, 2014.

[12] S. Günnemann, N. Günnemann, and C. Faloutsos. Detecting anomalies in dynamic rating data: A robust probabilistic model for rating evolution. In *KDD*, pages 841–850, 2014.

[13] D. Hernández, R. Guzmán, M. Móntes y Gomez, and P. Rosso. Using pu-learning to detect deceptive opinion spam. In *Proc. of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 38–45, 2013.

[14] B. Hooi, N. Shah, A. Beutel, S. Gunneman, L. Akoglu, M. Kumar, D. Makhija, and C. Faloutsos. Birdnest: Bayesian inference for ratings-fraud detection. *arXiv preprint arXiv:1511.06030*, 2015.

[15] X. Jin, C. Lin, J. Luo, and J. Han. A data mining-based spam detection system for social media networks. *Proceedings of the VLDB Endowment*, 4(12):1458–1461, 2011.

[16] N. Jindal and B. Liu. Opinion spam and analysis. In *WSDM*, pages 219–230, 2008.

[17] P. Kolari, A. Java, T. Finin, T. Oates, and A. Joshi. Detecting spam blogs: A machine learning approach. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1351, 2006.

[18] G. Koutrika, F. A. Effendi, Z. Gyöngyi, P. Heymann, and H. Garcia-Molina. Combating spam in tagging systems. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, pages 57–64, 2007.

[19] R. Y. Lau, S. Liao, R. C.-W. Kwok, K. Xu, Y. Xia, and Y. Li. Text mining and probabilistic language modeling for online review spam detection. *ACM Transactions on Management Information Systems (TMIS)*, 2(4):25, 2011.

[20] F. Li, M. Huang, Y. Yang, and X. Zhu. Learning to identify review spam. In *IJCAI*, volume 22, page 2488, 2011.

[21] H. Li, Z. Chen, B. Liu, X. Wei, and J. Shao. Spotting fake reviews via collective positive-unlabeled learning. In *ICDM*, pages 899–904, 2014.

[22] H. Li, Z. Chen, A. Mukherjee, B. Liu, and J. Shao. Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. In *ICWSM*, pages 634–637, 2015.

[23] H. Li, G. Fei, S. Wang, B. Liu, W. Shao, A. Mukherjee, and J. Shao. Modeling review spam using temporal patterns and co-bursting behaviors. *arXiv preprint arXiv:1611.06625*, 2016.

[24] H. Li, B. Liu, A. Mukherjee, and J. Shao. Spotting fake reviews using positive-unlabeled learning. *Computación y Sistemas*, 18(3):290–299, 2014.

[25] H. Li, A. Mukherjee, B. Liu, R. Kornfield, and S. Emery. Detecting campaign promoters on twitter using markov random fields. In *ICDM*, pages 290–299, 2014.

[26] J. Li, M. Ott, C. Cardie, and E. H. Hovy. Towards a general rule for identifying deceptive opinion spam. In *ACL*, pages 1566–1576, 2014.

[27] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw. Detecting product review spammers using rating behaviors. In *CIKM*, pages 939–948, 2010.

[28] R. D. Malmgren, J. M. Hofman, L. A. N. Amaral, and D. J. Watts. Characterizing individual communication patterns. In *KDD*, pages 607–616, 2009.

[29] C. D. Manning, P. Raghavan, H. Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[30] R. Mihalcea and C. Strapparava. The lie detector: Explorations in the automatic recognition of deceptive language. In *ACL*, pages 309–312, 2009.

[31] A. Mukherjee, B. Liu, and N. S. Glance. Spotting fake reviewer groups in consumer reviews. In *WWW*, pages 191–200, 2012.

[32] A. Mukherjee, V. Venkataraman, B. Liu, and N. S. Glance. What yelp fake review filter might be doing? In *ICWSM*, 2013.

[33] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *ACL*, pages 309–319, 2011.

[34] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[35] S. Rayana and L. Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *KDD*, pages 985–994, 2015.

[36] Y. Ren, D. Ji, and H. Zhang. Positive unlabeled learning for deceptive reviews detection. In *EMNLP*, pages 488–498, 2014.

[37] K. Santosh and A. Mukherjee. On the temporal dynamics of opinion spamming - case studies on yelp. In *WWW*, 2016.

[38] M. Steinbach, G. Karypis, V. Kumar, et al. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston, 2000.

[39] G. Wang, S. Xie, B. Liu, and P. S. Yu. Review graph based online store review spammer detection. In *ICDM*, pages 1242–1247, 2011.

[40] S. Xie, G. Wang, S. Lin, and P. S. Yu. Review spam detection via temporal pattern discovery. In *KDD*, pages 823–831, 2012.

[41] C. Xu and J. Zhang. Towards collusive fraud detection in online reviews. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 1051–1056. IEEE, 2015.

[42] C. Xu, J. Zhang, K. Chang, and C. Long. Uncovering collusive spammers in chinese review websites. In *CIKM*, pages 979–988, 2013.

[43] J. Ye and L. Akoglu. Discovering opinion spammer groups by network footprints. In *ECML/PKDD*, pages 267–282, 2015.

[44] J. Ye, S. Kumar, and L. Akoglu. Temporal opinion spam detection by multivariate indicative signals. In *ICWSM*, 2016.

[45] D. Yu, Y. Tyshchuk, H. Ji, and W. Wallace. Detecting deceptive groups using conversations and network analysis. In *ACL*, pages 26–31, 2015.

[46] L. Zhou, Y. Shi, and D. Zhang. A statistical language modeling approach to online deception detection. *IEEE Trans. Knowl. Data Eng.*, 20(8):1077–1081, 2008.