Proceedings of the Tenth International AAAI Conference on
Web and Social Media (ICWSM 2016)

Temporal Opinion Spam Detection by Multivariate Indicative Signals

Junting Ye, Santhosh Kumar, Leman Akoglu
Stony Brook University
Department of Computer Science
{juyye, smanavasilak, leman}@cs.stonybrook.edu

Abstract

Online consumer reviews reflect the testimonials of real people, unlike e.g., ads. As such, they have critical impact on potential consumers, and indirectly on businesses. Problematically, such financial incentives have created a market for spammers to fabricate reviews to unjustly promote or demote businesses, activities known as opinion spam (Jindal and Liu 2008). Most existing work on this problem have formulations based on *static* review data, with respective techniques operating in an *offline* fashion. Spam campaigns, however, are intended to make most impact during their course. Abnormal events triggered by spammers' activities could be masked in the load of future events, which static analysis would fail to identify. In this work, we approach the opinion spam problem with a *temporal* formulation. Specifically, we monitor a list of carefully selected indicative signals of opinion spam over time and design *efficient* techniques to both detect *and* characterize abnormal events in *real-time*. Experiments on two different datasets show that our approach is fast, effective, and practical to be deployed in real-world systems.

Introduction

Online product reviews play important role for e-commerce. New customers tend to prefer products with higher ratings as previous buyers have "testified" that the products are good choices. Driven by such benefits, spam or fake reviews have become a prevalent problem, for which effective detection algorithms are greatly needed (Jindal and Liu 2008).

Some existing works employed supervised techniques by extracting features based on text, ratings, product meta-data, etc. (Jindal and Liu 2008; Feng, Banerjee, and Choi 2012). These methods have a key challenge as ground truth is extremely hard to obtain (Ott et al. 2011). Meanwhile, unsupervised approaches have also been explored, which leverage linguistic (Ott et al. 2011), relational (Akoglu, Chandy, and Faloutsos 2013; Ye and Akoglu 2015), and behavioral clues (Mukherjee et al. 2013). Recently a unifying approach is also proposed to harness all these information sources, which outperforms any individual one (Rayana and Akoglu 2015). Most previous work formulate the opinion spam problem on *static* data, thus operate in an *offline* fashion (Fei et al. 2013; Li et al. 2015). (Xie et al. 2012) proposed an approach that utilizes temporal features, but it is specifically designed for detecting spam reviewers with single reviews.

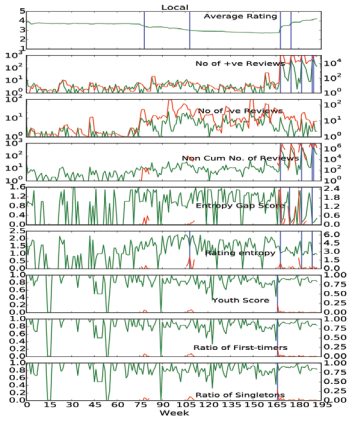

Figure 1: Time series of 9 indicative signals for an app (green: signal values, red: anomaly score, blue bars: anomalous time points). Our method detects 4 weeks of attacks that increase the average rating of a product with generally declining rating. Results discussed further in experiments.

In this work, we consider opinion spam as a *temporal* phenomenon (see Figure 1). Our key insight is that the spammers' activities trigger abrupt changes in the underlying generating processes of the data *at the time* of their operations. Such changes may later be reverted, or the impact of such events may be lost in the abundance of future events. As a result, an efficient real-time detector becomes essential.

For reproducibility, we share our source code[1] and full-length paper online (Ye, Kumar, and Akoglu 2016).

Indicative Signals of Opinion Spam

We consider the review data of a product p as a stream of review units ordered in time, $U_p = \{u_1, u_2, ..., u_i, ...\}$. Each unit consists of user ID, timestamp, review text, and rating; i.e, $u_i = (UID_i, t_i, c_i, r_i)$, where $t_i \leq t_j$ if $i < j$. We divide the timestamps into fixed-length time windows (ΔT) where we compute each time series based on data from these

[1]Source code of the proposed detector: http://www3.cs.stonybrook.edu/~juyye/code/ICWSM16_Code.zip

intervals. We identify eight types of indicative signals, and extract their time series for each product p.

1. **Average Rating**: This time series tracks the evolution of *cumulative* average rating. Let $\bar{U}_p^t = \{u_k | t_k \in [0, t*\Delta T)\}$ denote the set of p's reviews until the end of time window t, where $|\bar{U}_p^t| = m$. Then, $R_p^t = \frac{1}{m}\sum_{k=1}^{m} r_k$.
2. **Number of Reviews**: This time series tracks the total number of reviews within each time interval. Let $U_p^t = \{u_k | t_k \in [(t-1)*\Delta T, t*\Delta T)\}$ denote p's reviews within window t. Then, $C_p^t = \left|U_p^t\right|$.
3. **Number of Positive/Negative Reviews**: We also track the positive and negative review counts, as fake reviews have skewed ratings. $+C_p^t = \left|\{u_k | u_k \in U_p^t, r_k \in \{4,5\}\}\right|$, $-C_p^t = \left|\{u_k | u_k \in U_p^t, r_k \in \{1,2\}\}\right|$.
4. **Rating Entropy**: We monitor rating entropy over time. Let p_r^t denote the fraction of reviews with rating value equal to r in window t. Then, $E_p^t = -\sum_{r=1}^{5} p_r^t \cdot \log p_r^t$.
5. **Ratio of Singletons**: We track the ratio of one-time reviewers since fake reviews could be posted by newly created accounts. $S_p^t = \frac{|U_s^t|}{C_p^t}$, where U_s^t is the user set who posted their *only* review (to p) during window t.
6. **Ratio of First-timers**: Spammers also may target multiple products simultaneously. Let U_f^t be the set of p's reviewers who posted their first but not necessarily only review during window t, then $F_p^t = \frac{|U_f^t|}{C_p^t}$.
7. **Youth Score**: We compute the age of the reviewer UID_k at the time they posted u_k for p, by $A_k = t_k - t_0^{UID_k}$, where $t_0^{UID_k}$ is the time at which reviewer UID_k posted their first review. The youth score is the average of reviewer ages at the time they posted for p, i.e. $Y_p^t = \frac{1}{C_p^t}\sum_{u_k \in U_p^t} 2 \cdot (1 - \frac{1}{1+\exp(-A_k)})$.
8. **Temporal Gap Entropy**: This series tracks the time-gap between consecutive reviews $\{u_k, u_{k+1}\} \in U_p^t$ of p within window t, creates a histogram, and computes the entropy. $G_p^t = -\sum_{b=1}^{\lceil log_2 \Delta T \rceil + 1} p_b^t \cdot \log p_b^t$, where p_b^t is the fraction of gaps in logarithmically-growing bin b in t.

Temporal Opinion Spam Detection

Our temporal approach to opinion spam detection consists of four main steps: *(i)* extracting temporal signal values as defined in the previous section; *(ii)* detecting changes in what is called the *lead* signal; *(iii)* checking whether we also observe temporal anomalies in the *supporting* signals; and *(iv)* ranking targeted products based on the number and magnitude of anomalies found in their timeline. In the following subsections, we present the details of steps *(ii)-(iv)*.

Anomalies in the Lead Signal

Of all the indicative signals extracted over time, we dedicate one of them as the *lead* signal. The lead can be chosen as the measure that spammers particularly aim to manipulate (e.g., avg. rating), or a measure for which spamming activities could trigger a change (e.g., positive review count).

For anomaly detection on the lead time series, one can use any detection algorithm that provides real-time capability. One important issue is the semantics of the lead signal. Average rating is cumulative, and our goal is to find *change points* in both directions (either increase or decrease). For this lead, we use the cumulative sum (CUSUM) change detection algorithm (Page 1954). On the other hand, we track the non-cumulative number of positive and negative reviews per time window ΔT, with a goal to spot *anomalies* in the form of bursts (i.e., large increases). For such leads, we use the autoregressive (AR) model for modeling the lead series, and use the deviation from the forecasted value as the anomaly score s at a new time point.

For change/anomaly detection, choice of a threshold is critical to flag alerts. For a given (lead) signal and a detector, we maintain the distribution $\mathcal{D}(S|T,P)$ of the anomalousness scores S; (i) across time points T and (ii) across products P. We then employ Cantelli's inequality[2] to identify a theoretical threshold $\delta = (\mathcal{D}, \eta)$, where η is the expected percentage of anomalies. For a given score s_p^t for product p at time t, we flag an alert if $s_p^t > \delta$ and the anomaly is in the direction of what a suspicious activity would create (i.e., increase or decrease in the corresponding signal).

Anomalies in the Supporting Signals

Lead signal alone is not sufficient to indicate the occurrence of spamming activities. Therefore, we investigate further the supporting signals, to verify if "alarms" triggered by the lead signal are indeed the consequences of spamming activities.

In order to detect anomalies in each supporting signal, we propose LOCALAR, which only focuses on and scores the time points around the "alarms" produced by the lead signal, and hence selectively ignores the other time points. This reduces the time complexity to *sublinear*, in terms of the total number of time points. In the following, we elaborate on the steps of LOCALAR, which is shown in Algorithm 1.

At time t_i, we check if there is a close-by "alarm" t_a in the lead signal. If not, then we exit the algorithm (Lines 3-4). This step accelerates the algorithm significantly in two aspects: (i) it skips anomaly score computation for points away from the lead "alarms", as a result of which (ii) feature extraction for a large body of time points in supporting signals can also be skipped. Next, we select a proper (integer) k that minimizes the total square error over a window of L values[3] before v_i (Lines 5-12). Specifically, we pick a candidate k' and initialize the square error sum $S_{k'}$ to 0. We compute the square error s_{i-j} between the inputs v_{i-j} and predictions $\hat{v}_{i-j}$ of an AR model of order k', for all L values before v_i. Sum of square errors is denoted by $S_{k'}$. We then choose the k' with the minimum $S_{k'}$ as the order of our AR model at time t_i. As temporal dependence drops by distance in time, we focus on small $k' \in [1,5]$. Through Lines 13-18, we carefully examine the time points around t_a. We compute the

[2]Unlike the t-test which assumes Gaussian data, Cantelli's inequality does not make any distributional assumptions on $\mathcal{D}$.

[3]Window size L is essentially the training data size used to estimate k. Choice of L poses a trade-off between the estimation quality and running time. In experiments we find $L = 8$ effective.

Algorithm 1: LOCALAR

Input: t_i, t_a, L, v^i_{i-L-5}, δ
Output: $s^i_{t_a-2}$, $O^i_{t_a-2}$
if $t_i - t_a > 2$ **then** *// exit if no recent anomaly in lead signal*
  Exit
foreach $k' \in [1, 5]$ **do** *// select k that minimizes square error*
  Init $S_{k'} = 0$
  foreach $j \in [1, L]$ **do**
    $\boldsymbol{\theta} = AR(v^{i-j-1}_{i-j-k'}, k')$ (where $\boldsymbol{\theta} = \{\boldsymbol{\omega}, \mu, \sigma\}$)
    $\hat{v}_{i-j} = \boldsymbol{\omega}(v^{i-j-1}_{i-j-k'} - \boldsymbol{\mu}) + \mu$
    $s_{i-j} = (v_{i-j} - \hat{v}_{i-j})^2$
    $S_{k'} = S_{k'} + s_{i-j}$
$k = k'_{min}$, where $\forall k' \in [1, 5], S_{k'} \geq S_{k'_{min}}$
foreach $t_j \in [t_a - 2, t_i]$ **do** *// check time points around t_a*
  $\boldsymbol{\theta_j} = AR(v^{j-1}_{j-k}, k)$
  $\hat{v}_j = \boldsymbol{\omega_j}(v^{j-1}_{j-k} - \boldsymbol{\mu_j}) + \mu_j$
  $s_j = (v_j - \hat{v}_j)^2$
  if $s_j > \delta$ *& SemSus(v_j) is* $True$ **then**
    $O_j = 1$

square error for values at and before v_i at this step, using the estimated k. If the square error is larger than the anomaly threshold and $SemSus(v_j)$ returns $True$[4], then output value O_j is assigned as 1, i.e. anomalous.

Scoring and Ranking Products

We propose a function to score products by suspiciousness based on *(i)* the number and *(ii)* magnitude of the temporal anomalies among its indicative signals. Four measures are designed to quantify product suspiciousness.

First is the fraction of anomalies among product p_i's 9 indicative signals at t_j. That is, $f_1(p_i, t_j) = \sum_{l=1}^{9} O^{(l)}_{j,p_i}/9$, where $O^{(l)}_{j,p_i} \in \{0, 1\}$ is the anomaly label of signal l of product p_i at t_j. The second and third measures are respectively the average and maximum magnitude of the anomalies, and can be written as $f_2(p_i, t_j) = \sum_{l=1}^{9} s^{(l)}_{j,p_i} / \sum_{l=1}^{9} O^{(l)}_{j,p_i}$ and $f_3(p_i, t_j) = \max_{l=1\ldots9} s^{(l)}_{j,p_i}$, where $s^{(l)}_{j,p_i}$ is the anomaly score of signal l of product p_i at t_j. Finally, $f_4(p_i, t_j) = \sum_{l=1}^{9} w_l \cdot s^{(l)}_{j,p_i}$ is the weighted sum of the anomaly scores, where $w_l = 1/\sum_{t=1}^{j} O^{(l)}_{t,p_i}$ which is inversely proportional to the number of anomalies in signal l.

We then use the empirical CDF to normalize the feature values, i.e. $F_g(p_i, t_j) = P(f_g \leq f_g(p_i, t_j)), \quad g = 1, \ldots, 4$, where $F_g(p_i, t_j)$ is the fraction of f_g's that are smaller than or equal to $f_g(p_i, t_j)$ across all products and all time points before t_j. The larger the $F_g(p_i, t_j)$, the larger the anomalousness. Finally, we compute p_i's suspiciousness score $A(p_i, t_j)$ at time t_j as an average of the $F_g(p_i, t_j)$'s.

[4] v_j is "semantically suspicious" (denoted as $SemSus(v_j) = True$) if either *(i)* $v_j > v_{j-1}$ and large value indicates spamming activities (e.g., ratio of singletons), or *(ii)* if $v_j < v_{j-1}$ and small value indicates suspicious behaviors (e.g., rating entropy).

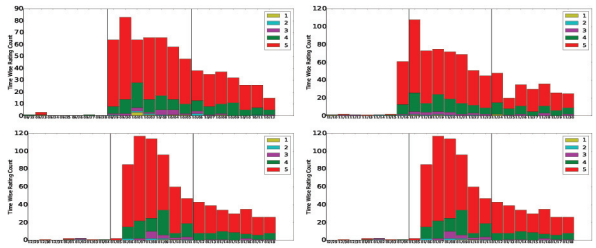

Figure 2: Stacked bars show daily review counts for 4 detected campaigns in Figure 1 (week before, during, and after campaign, separated by vertical bars). Stacks represent counts for ratings 1-5. Notice that spam campaigns involve mostly 5-star reviews (hence the bumpy increase in average rating week by week after each campaign).

Experiments

Datasets

SoftWare Marketplace (SWM) consists of reviews for all software products (apps) from the entertainment category in a popular online software marketplace. It contains 15,094 apps with over 1.1 million reviews by over 966,000 users, and spans 198 weeks between July 2008 and April 2012.

FLIPKART contains reviews from flipkart.com, an e-commerce site which provides a platform for sellers to market products to customers. It contains about 545,000 products with roughly 3.3 million reviews by 1.1 million users, and spans 180 weeks between August 2011 to January 2015.

Results

We manually inspect the top-ranked products from both datasets, and provide evidence through case studies.

SWM Case I: Game app This app (see Figure 1), which started off with an average rating of 4, declined below 3-stars between weeks 75 to 165. A series of spam campaigns are then executed in weeks 168, 175, 182, and 189 (notice how these campaigns are organized every 7 weeks—a strong indication of manipulation). When we use 'Average Rating' as the lead signal, we spot the first two campaigns, whereas when 'Num. of + Reviews' is used as the lead, all 4 weeks are detected. Nearly all the supporting signals also change simultaneously. Figure 2 shows the daily review counts for the week before, at, and after the spam campaigns, for each of the 4 detected campaigns, along with the distribution of ratings per day. Most reviews are from *singletons* that rated this app with 4 or 5 stars. We also find that singleton reviews have duplicates or near duplicates. For example all the following text snippets appear multiple times across different reviews of this app: "Great app for gamers", "Great App For Gaming news", "Easy to use", "Must have app for gamers".

FLIPKART Case I This product is detected as anomalous in week 35 (Figure 3 left). During this period, over 80 reviews are written. Most reviews are rated 3- or 4-stars, but only a few 5-stars, while being able to increase average rating. These mixed ratings appear to be for better camouflage. Moreover, most reviewers are non-singletons (unlike in SWM) although

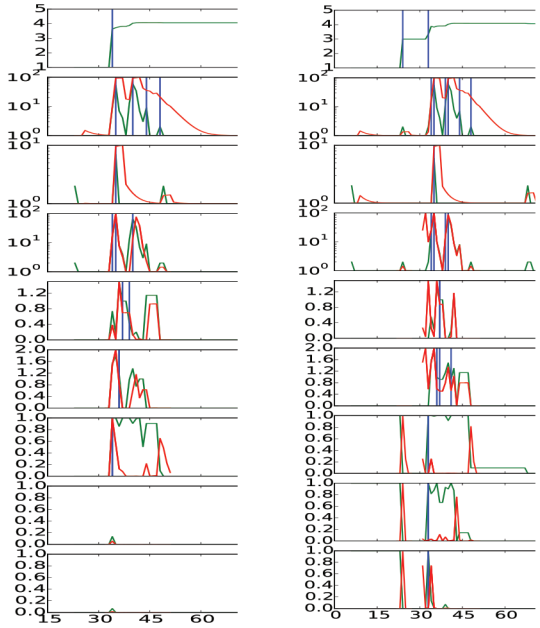

Figure 3: Partial time series for 9 indicative signals for two different products from FLIPKART that were spammed by same reviewers during same time periods (week 35 and 40).

they are young accounts. This suggests that other products might also have been spammed by the same reviewers. We confirmed this conjecture by finding that a list of other products (one of which is shown in Figure 3 right) are rated similarly by these same users *during the same time period* (these periods are also detected as anomalous by our method). Further, we find that all these products are hair-related, including straighteners, dryers, and shavers, potentially belonging to the same seller.

FLIPKART Case II Our second case study from FLIPKART is for an anomalous book, whose average rating increased to 4.4 on week 95 (see Figure 4). We find that this book received 125 5-star reviews in mainly two days during that week. Surprisingly those were from non-singletons, who also reviewed another product—also a book (!) Further investigation revealed that those were 2 out of 3 books of an author. Both books have average ratings ∼4.4, while they are rated ∼3.3 on another review system, Goodreads.com. Moreover, we found that almost all 125 reviewers have similar behavioral pattern: 5-star reviews written 7 PM–11PM on June 8 and 11AM–7PM on June 9, 2013. What is more, their reviews follow nearly the *same order* for both books.

For more details on these case studies as well as additional ones, we refer to (Ye, Kumar, and Akoglu 2016).

Conclusion

Opinion spam has become a prevalent problem, for which a vast body of methods operate in an offline fashion on a collection of static data. In this work, we brought emphasis to the aspect of *time*, and approached this problem with a novel temporal formulation. We proposed a new methodology that

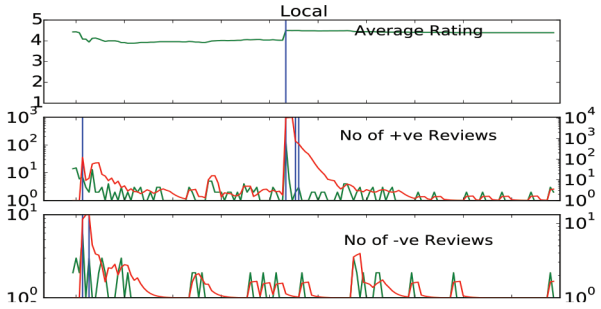

Figure 4: Series for 3 lead signals for a FLIPKART book.

(i) monitors a comprehensive list of indicative signals of spam over time, *(ii)* spots anomalous events in real-time, and *(iii)* provides pointers to specific time intervals for manual inspection and characterization. As such, our approach exhibits desirable properties, as it is online, efficient, and descriptive. Importantly, our method is general enough to be employed for other applications in which multiple signals are monitored over time, such as enterprise security, environmental monitoring, and surveillance systems.

Acknowledgments. Research is supported by the NSF CAREER 1452425, IIS 1408287, the DARPA Transparent Computing Program under Contract No. FA8650-15-C-7561, a Facebook Faculty Gift, and an R&D grant from Northrop Grumman. Any conclusions expressed in this material are of the authors and do not necessarily reflect the views, expressed or implied, of the funding parties

References

Akoglu, L.; Chandy, R.; and Faloutsos, C. 2013. Opinion fraud detection in online reviews by network effects. In *ICWSM*.

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

Feng, S.; Banerjee, R.; and Choi, Y. 2012. Syntactic stylometry for deception detection. In *ACL*.

Jindal, N., and Liu, B. 2008. Opinion spam and analysis. In *WSDM*.

Li, H.; Chen, Z.; Mukherjee, A.; Liu, B.; and Shao, J. 2015. Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. In *ICWSM*.

Mukherjee, A.; Kumar, A.; Liu, B.; Wang, J.; Hsu, M.; Castellanos, M.; and Ghosh, R. 2013. Spotting opinion spammers using behavioral footprints. In *KDD*.

Ott, M.; Choi, Y.; Cardie, C.; and Hancock, J. T. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *ACL*.

Page, E. S. 1954. Continuous Inspection Schemes. *Biometrika* 41.

Rayana, S., and Akoglu, L. 2015. Collective opinion spam detection: Bridging review networks and metadata. In *KDD*.

Xie, S.; Wang, G.; Lin, S.; and Yu, P. S. 2012. Review spam detection via temporal pattern discovery. In *KDD*.

Ye, J., and Akoglu, L. 2015. Discovering opinion spammer groups by network footprints. In *ECML/PKDD*.

Ye, J.; Kumar, S.; and Akoglu, L. 2016. *Temporal Opinion Spam Detection by Multivariate Indicative Signals*. http://arxiv.org/abs/1603.01929.