ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/254464446

Review spam detection via temporal pattern discovery

Article · August 2012

DOI: 10.1145/2339530.2339662

CITATIONS
99

READS
122

4 authors:

Sihong Xie
Lehigh University
29 PUBLICATIONS **473** CITATIONS
SEE PROFILE

Guan Wang
University of Illinois at Chicago
12 PUBLICATIONS **359** CITATIONS
SEE PROFILE

Shuyang Lin
University of Illinois at Chicago
11 PUBLICATIONS **146** CITATIONS
SEE PROFILE

Philip S. Yu
University of Illinois at Chicago
1,207 PUBLICATIONS **53,415** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Product Compatibility Analysis View project

Project Detecting Deception in Twitter View project

All content following this page was uploaded by Guan Wang on 11 November 2014.

The user has requested enhancement of the downloaded file.

Review Spam Detection via Temporal Pattern Discovery

Sihong Xie[†] Guan Wang[†] Shuyang Lin[†] Philip S. Yu[†‡]
[†] Department of Computer Science, University of Illinois at Chicago, Chicago, IL
[‡] Computer Science Department King Abdulaziz University Jeddah, Saudi Arabia
{sxie6, gwang26, slin38, psyu}@uic.edu

ABSTRACT

Online reviews play a crucial role in today's electronic commerce. It is desirable for a customer to read reviews of products or stores before making the decision of what or from where to buy. Due to the pervasive spam reviews, customers can be misled to buy low-quality products, while decent stores can be defamed by malicious reviews. We observe that, in reality, a great portion ($> 90\%$ in the data we study) of the reviewers write only one review (singleton review). These reviews are so enormous in number that they can almost determine a store's rating and impression. However, existing methods did not examine this larger part of the reviews. Are most of these singleton reviews truthful ones? If not, how to detect spam reviews in singleton reviews? We call this problem *singleton review spam detection*.

To address this problem, we observe that the normal reviewers' arrival pattern is stable and uncorrelated to their rating pattern temporally. In contrast, spam attacks are usually bursty and either positively or negatively correlated to the rating. Thus, we propose to detect such attacks via unusually correlated temporal patterns. We identify and construct multidimensional time series based on aggregate statistics, in order to depict and mine such correlations. In this way, the singleton review spam detection problem is mapped to a abnormally correlated pattern detection problem. We propose a hierarchical algorithm to robustly detect the time windows where such attacks are likely to have happened. The algorithm also pinpoints such windows in different time resolutions to facilitate faster human inspection. Experimental results show that the proposed method is effective in detecting singleton review attacks. We discover that singleton review is a significant source of spam reviews and largely affects the ratings of online stores.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
KDD'12, August 12–16, 2012, Beijing, China.
Copyright 2012 ACM 978-1-4503-1462-6 /12/08 ...$10.00.

General Terms

Algorithms, Experimentation

Keywords

Review spam, time series, adversarial data mining

1. INTRODUCTION

Online reviews and ratings about products and stores are essential parts in today's electronic commerce where they provide helpful information for potential customers. A product or store with a decent rating and a high proportion of positive reviews will attract more customers and larger amount of business, while a couple of negative reviews/ratings could substantially harm the reputation, leading to financial losses. Since there is no rule governing online reviews and ratings, some product providers or retailers are leveraging such public media to defame competitors and promote themselves unfairly, or even to cover the truth disclosed by genuine reviews. For example, suppose a customer finds the delivery service of a certain store unacceptably slow, she then writes a review about the fact and gives it a low rating on a review website. This review and rating present a unfavourable impression of the store to potential customers, who might choose other stores after reading that review. In order to avoid the drainage of business caused by this negative yet truthful review, the store could employ or entice a group of people to write undeserving positive reviews about the delivery service. Similarly, the store could also ask these people to write unfavorable reviews about its competitors, from which the store would like to distract customers. These hired reviewers are called spammers and the reviews they write are called spam reviews. In order to protect customers, honest online stores and the whole electronic commerce environment, it is desirable to detect spam reviews and take proper measures.

Previous works propose to use features of review contents and reviewers' behaviors [4, 9, 10, 1] or graph connecting reviewers, stores and reviews [5], to detect spam reviews. These methods work best in the situations where spammers write many reviews (see related work). In reality, however, most reviewers write only one review. For example, 68% of the reviewers write a single review in the Amazon review dataset studied in [9], and this percentage is 90% in the dataset we study here. If a review is the only review a reviewer has written, we call it a *singleton review* (SR for short). In fact, as we shall see later, the SR definition can be generalized to cover reviewers with a few reviews, not necessarily just one. One question is: are most of these SRs honest reviews? The answer is probably not, due to the *nature* of spam attacks. For a store to rapidly raise its fame and rating (resp. defame others) it is desirable to have spammers post plenty of favorable (resp. unfavorable) reviews about it (resp. its competitors) in a short time. Usually a spammer would not post many reviews with similar ratings for a store under the same name. Instead, he would rather write spam reviews under different names to avoid being caught. This spamming strategy brings a large number of SR. Most of the statistics adopted by previous works would not work on such singleton reviews. For example, the mean and standard deviation of ratings given by a reviewer [4] become meaningless if this reviewer has written only one review; in the rule or frequent pattern based detection method [10], singleton reviews are also ignored due to their low significance.

In the present paper, we focus on singleton review (SR) spam attack detection, and "spammer" refers to "SR spammer" if not otherwise specified. According to the above observations, we propose a novel approach that maps the SR spam detection problem to a abnoramlly correlated temporal pattern detection problem. The proposed algorithm is based on multi-scale multidimensional time series anomaly detection. In particular, we construct statistics whose joint anomaly could be a strong indicator of SR spam attacks. The identified statistics includes the average rating, the total number of reviews, and the ratio of singleton reviews among all reviews. We collect these statistics from ratings and reviews for each store to build the multidimensional time series, base on which we develop an SR spam detection model. We combine temporal curve fitting and LCS (Longest Common Sub-sequence) algorithms to find out abnormal sections in each dimension of the time series. We then devise a ranking-based algorithm to consolidate the anomalies in all dimensions to find out temporally correlated abnormal sections. Furthermore, since short term fluctuations are common in the constructed time series, we start with a larger time window (e.g. 2 months) to smooth out such noisy changes of the time series, so that any significant abnormal pattern can be detected robustly, therefore reducing false positive rate. After any singleton review attack is detected, we scale down the window size, so that the exact abnormal points become more obvious and one can quickly locate the suspicious reviews.

This paper makes the following contributions:

- We identify and formulate the singleton spam review detection problem, which has not been addressed in previous works. This problem is important yet difficult to solve, as we will point out in Section 2.
- We develop a new principle for SR spam detection based on the observation that the arrival pattern of SR tends to be bursty (Section 3.1) and temporally correlated to the rating. We identify multiple temporal statistics of reviews to support detection based on this principle. Instead of following traditional approaches, which consider the reviews as a static collection of documents, here we take a new approach of treating them as a series of events.
- We propose a hierarchical detection criterion to detect SR spam attacks robustly and accurately. This feature is especially useful for online review website quality and trust monitoring.
- We find the number of singleton spam reviews to far exceed the number of other types of spam reviews identified in the literature.

2. MOTIVATION OF SINGLETON REVIEW SPAM DETECTION

In the data studied here and in [9], we can observe that the reviewers who write only one review (a singleton review) dominate the body of reviewers. The sheer number of these reviews implies that the rating of a store or product, and potentially the customers' choices can be greatly manipulated by these reviews. This property make SR particularly attractive to spammers. Despite the significance of detecting SR spams, no existing work has addressed this problem. Indeed, detecting SR spams can be challenging. If a reviewer has written only one review, simply looking at this singleton review reveals little information about the true intention behind it, so it is hard for machines or even human beings to draw a conclusion. For example, in the experiment, we find that one reviewer said "For the few times that I've contacted customer service via phone, email, or chat, the person has always been helpful and gone out of his/her way." At a first glance, this is a normal review talking about customer service. Since it is the only review the reviewer has written, existing spam detection algorithms will simply ignore this review. Even for human beings, it is extremely difficult to tell if this is a spam review or not. However, if we look at the aggregate reviewers' behaviors in a temporal way, we can find that this review was written in a period when there was a burst of SRs and the rating of the store went up dramatically. It is abnormal for the number of reviews, the ratio of SR and the store rating to be temporally correlated, this review is pretty suspicious.

In particular, spammers have to write many positive (or negative) reviews in a short time, otherwise, the spammers are not effective in promoting or defaming a store or product. However, if a spammer posts his reviews quickly under the same name, he can be easily detected by checking the duration between two consecutive reviews with similar rating from a single person. So writing spam reviews under different names is a safer way. Based on the above reasoning, we make the following conjectures on SR spam attacks. When such an attack occurs in a certain period, there tends to be a sharp increase in the number reviews and the ratio of SRs, together with an increase (or decrease) in the average rating. Therefore, we can transform the SR spam detection problem to abnormally correlated temporal pattern detection in a multidimensional time series consisting of the above three indices. Note that spammers may want to evade the proposed method by writing more than one but not too many reviews. For these spammers, we can easily modify the algorithm to catch them (see Section 3.2.1) and we focus on the detecting SR spams. In the next section, we make several assumptions about reviewers' arrival patterns, which define a necessary condition of SR spam attacks.

3. SINGLETON REVIEW SPAM DETECTION MODEL

3.1 The Model of Reviewer Behavior

It is the difference in spammers' and genuine reviewers' behaviors that motivates the proposed algorithm. We show

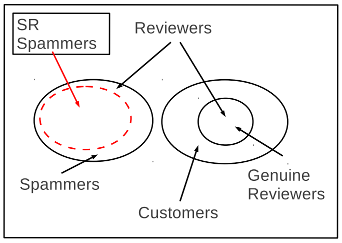


Figure 1: Relationship of Spammers, Reviewers and Customers

the relationship of reviewers, customers, spammers and SR spammers in Figure 1. Note that the areas do not reflect the actual number of groups of people. On one hand, we define a reviewer as a name under which at least one review is written. A customer is a person who has shopping experience with at least one store, and becomes a reviewer if he/she contributes a review. On the other hand, there are two categories of spammers, namely, SR spammers and other spammers. We assume a customer cannot be a spammer (the customer set does not intersect the spammer set in Figure 1). Indeed, though it is possible for a store to entice its customers to write favorable reviews for it, the store has to take the risks of the disclosing of such dishonesty to the public by its customers and harming the store's reputation. Instead, a store is more willing to hire dedicated spammers. Also, a spammer might later become a customer and write a genuine review, but our aim here is to detect *any* spamming activities, instead of telling spammers from customers. Therefore, we assume that a customer cannot be a spammer.

We make certain assumptions of reviewers' behaviors, divided into two phases: the arrival and writing phase. In the arrival phase, a customer buys something from a store or a spammer is hired or enticed by a store to write fake reviews. The writing phase is when a reviewer writes a review. There are mainly three patterns of arrival phase behaviors: normal arrival, promotion/sale event arrival, and spam attack pattern. First, the normal arrival pattern can be modeled by a homogeneous Poisson process with a fixed rate λ. A Poisson process is a set of random variables $\{N(t) : t \geq 0\}$ satisfying the following properties [8]:

- $\Pr\{N(t+h) - N(t) = 1 | N(t) = n\} = \lambda h + o(h)$ as $h \to 0$, for $n = 0, 1, \ldots$
- $\Pr\{N(t+h) - N(t) = 0 | N(t) = n\} = 1 - \lambda h + o(h)$ as $h \to 0$
- $N(0) = 0$

where $N(t)$ is the number of arrivals up to time t. λ is a constant controlling the intensity of arrivals, with a larger λ indicating more arrivals in a unit of time. Second, it is possible for a store to promote their products over a period and therefore increase the traffic of customers and reviews. We model this arrival pattern using a non-homogeneous Poisson process, with the rate parameter being a function of time $\lambda(t)$. Third, the spam attack arrival pattern is pretty much like that in the promotion mode, since a large number of spammers would be hired or enticed by the store.

In the writing phase, we model the writing behaviors of normal reviewers and spammers. First, in order to get the rewards offered by the store that tries to commit SR spam attacks, spammers tend to write spam reviews in a hurry, and there is seldom a delay in posting spam reviews. Therefore, we assume that the time when a spammer writes a review is the same as the time she arrives, and the spam reviews' arrival pattern is the same as spammers' arrival pattern, a bursty one. Second, for genuine reviewers, we claim that there are some random factors associated with the delays in posting their reviews after their shopping experiences, by the following reasons. A genuine reviewer seldom writes a review right after she shops with a store. Instead, most of them would write their reviews after receiving and trying out the products for some time. Therefore, one random factor is the time spent on delivery, this factor depends on how a customer and a store choose the way of delivery, the traffic and logistic conditions and so on. Another random factor is the time spent on tryouts, which depends on customers' personal behaviors. The randomness associated with the delay in a genuine reviewer's posting a review smoothes out the arrival intensity of reviews, even in a promotion event. In orther words, their reviews are less likely to concentrate in a short period and causing bursty peaks.

According to the above analysis, a spam attack tends to create a burst on the review arrival process, which is distinct from the normal and even promotion review arrivals. Nonetheless, as fluctuations in the volume of reviews do exist, bursty patterns in review arrival do not necessarily imply SR spam attacks. Observe that spammers are brought together to bring up or down the rating of a store, the ratings of spam reviews are more likely to correlate with these reviews' arrivals. In contrast, because genuine reviewers' opinions about a store vary wildly, depending on their satisfaction with speed of delivery, quality of products and customer services, etc. If we average the ratings of genuine reviews in a certain period of reasonable length, the positive and negative ratings will cancel out each other, therefore, the average ratings should be stable over time and independent of genuine reviews' arrivals. In summary, we should look at the joint abnormal patterns in review arrival and rating to detect such attacks more robustly.

3.2 A Correlated Temporal Anomalies Discovery based Approach

According to the above assumptions, we propose an SR spam detection approach based on correlated abnormal patterns discovery in multidimensional time series. To raise or lower the rating of a store safely and rapidly, spammers tend to post a large number of reviews with a high or low rating under different names. If there is a sharp increase in the volume of (singleton) reviews with rating increases or decreases dramatically at the same time, it is highly likely that the rating is manipulated by the newly arrived reviews. Therefore, detection by exploiting the correlation between ratings and volume of (singleton) reviews is more robust compared to using any single time series. In the paper, we will focus on the case where the rating goes up dramatically. The case with rating going down dramatically can be similarly modelled.

In the remaining of this section, we first describe how to construct multidimensional time series capturing reviewers' behaviors (Section 3.2.1). Then in Section 3.2.2 we describe algorithms to detect correlated anomalies in all three time series. Lastly, in Section 3.2.3, we describe the proposed hierarchical framework for robust SR spam detection.

3.2.1 *Time Series Construction*

The detection approach is based on time series of the number of reviews, average ratings and the ratio of singleton reviews. The data we study here is a set of reviews with texts and ratings posted for different stores on a review website in a certain time period. To construct these time series, we discard text information and keep the posting time and ratings of the reviews. This is reasonable as there exist other spam detection algorithms utilizing text information, so they are complementary methods to the proposed algorithm. The resulting data can be seen in this way: each store s has a series of ratings sorted in ascending order of posting time.

$$R(s) = \{r_1, \dots, r_{n_s}\}$$

$$TS(s) = \{ts_1, \dots, ts_{n_s}\}$$

where n_s is the number of reviews for store s, and ts_i is the time stamp when r_i is written, $ts_i \leq ts_j$ for all $1 \leq i < j \leq n_s$. After choosing the time windows size (denoted by Δt), the time interval under investigation (denoted by $I = [t_0, t_0 + T]$) can be divided into $N = T/\Delta t$ consecutive time windows or sub-intervals. Each time window is of length Δt and contains reviews posted during that time window. Let I_n denote the n-th time window, so

$$I_n = [t_0 + (n-1)\Delta t, t_0 + n\Delta t], \qquad I = \bigcup_{n=1}^{N} I_n$$

Given a time window I_n, we compute the average rating f_1, the number of reviews f_2, and the ratio of singleton reviews f_3. Formally,

$$f_1(I_n) = \sum_{ts_j \in I_n} r_j / f_2(I_n)$$

$$f_2(I_n) = |\{r_j : ts_j \in I_n\}|$$

$$f_3(I_n) = |\{r_j : ts_j \in I_n, r_j \text{ comes from an SR}\}| / f_2(I_n)$$

where $|A|$ denotes the cardinality of the set A. Given a store s, time interval $I = [t_0, t_0 + T]$ and time window size Δt, these aggregate functions represent a three dimensional time series and can be collectively represented by

$$F_s(I, \Delta t) = \begin{bmatrix} f_1(1) & \dots & f_1(N) \\ f_2(1) & \dots & f_2(N) \\ f_3(1) & \dots & f_3(N) \end{bmatrix}_s$$

where $f_i(n)$ is a shorthand for $f_i(I_n)$, $i = 1, 2, 3$. In the following, we drop the index on stores and let $F(I, \Delta t)$ denote the time series constructed for a certain store. The way we construct these time series can be generalized to handle spammers who write just a few reviews with similar ratings. We can simply treat all the reviews as SR by ignoring reviewers' ids, then the way we construct these time series still makes sense and the proposed algorithm can detect SR attacks (see next section).

3.2.2 *Correlated Abnormal Patterns Detection in Multidimensional Time Series*

Given the three time series of a store, we would like to find out correlated abnormal blocks on all three series. In other words, these blocks should simultaneously present sudden increases in rating, ratio of singleton reviews and the number of reviews. Here we focus on the singleton review detection methodology based on burst detection algorithms. Instead of inventing a novel burst detection algorithm, which is not the focus in this paper, we use a three-step approach for the detection. First, on each dimension, we employ a Bayesian change point detection algorithm [3] to fit curves using the time series (other curve fitting algorithms will do the job, too). As an example, we plot the time series along with the fitted curves in Figure 2. We then apply a simple template matching algorithm to the fitted curves to detect bursty patterns. Lastly, a sliding window finds out the blocks in time series corresponding to a joint burst in all dimensions of the time series. In the above example, a joint burst is highlighted by the red box in Figure 2.

Assuming that we have obtained the fitted curves, we describe in what follows the last two steps in details, for the curve fitting algorithm, please refer to [3]. Let $C = \{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3\}$ be the fitted curves of the three dimensions of a time series. All curves have the same length (number of samples), which is also defined as the length of C. First, we want to detect sudden increases in each of the three curves separately. Based on the description of the arrival process in Section 3.1, this can be transformed to the problem of template matching. We use a step function-like template to represent a sudden rise in values

$$\mathbf{v} = \{-0.5, -0.5, 0.5, 0.5, 0.5\}$$

Note that one can use other values for $\mathbf{v}$ so long as it represents a sharp increase temporally. If a block on a fitted curve $\mathbf{c} = \{c_1, \dots, c_n\} \in C$ is found to “match” this template well, then we find an anomaly of interest on the curve. One can obtain all blocks of $\mathbf{c}$ by sliding a window through $\mathbf{c}$, and all points falling into the window form a block which is denoted by

$$\mathbf{b} = \{c_{i_1}, \dots, c_{i_5}\}$$

where $1 \leq i_k \leq n$ for $k = 1, \dots, 5$ and $i_k + 1 = i_{k+1}$ for $k = 1, \dots, 4$. Note that the length of a block is chosen to have the same length as the template. We use a modified longest common substring (LCS) for matching [12] between $\mathbf{v}$ and $\mathbf{b}$. In general, suppose we want to find the degree of match between two sequences $\mathbf{z}^1 = \{z_1^1, \dots, z_n^1\}$ and $\mathbf{z}^2 = \{z_1^2, \dots, z_n^2\}$. Without loss of generality, one can think of $\mathbf{z}^1$ as $\mathbf{v}$ and $\mathbf{z}^2$ as $\mathbf{b}$. In the modified LCS, how well two sequences match each other is measured by the number of points in one sequence matching those in the other sequence. By a “match” between two points, we mean the absolute difference between the values of two points is less than a given threshold ϵ. The modified LCS algorithm uses the following dynamic programming formula to find out how many matches occur between $\mathbf{z}^1$ and $\mathbf{z}^2$, for $0 \leq i, j \leq n$ and $|i - j| \leq 1$:

$$M(i,j) = \begin{cases} 0, & \text{if } i \text{ or } j = 0 \\ 1 + M(i-1, j-1), & \text{if } |z_i^1 - z_j^2| < \epsilon \\ \max\{M(i-1, j), \\ \qquad M(i, j-1)\}, & \text{otherwise} \end{cases}$$

where $M(i, j)$ records the number of matches between subsequences $\{z_1^1, \dots, z_i^1\}$ and $\{z_1^2, \dots, z_j^2\}$. The constraint $|i - j| \leq 1$ makes sure that, $z_i^1 \in \mathbf{z}^1$ is not matched to a point $z_j^2 \in \mathbf{z}^2$ far away from the position of z_i^1.

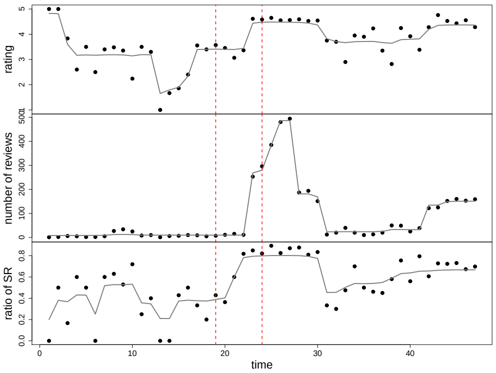


Figure 2: Bursty Patterns Detected in Store 24779

Algorithm 1 Bursts Detection in Single Time Series (**BD-STS**)

```
Input: fitted curve $\mathbf{c}$, template $\mathbf{v}$
Output: top $k$ ranked periods with bursty pattern
$m$ = length of $\mathbf{c}$.
$n$ = length of $\mathbf{v}$.
for $i = 1 \rightarrow m - n + 1$ do
    Normalize $\mathbf{c}[i : i + n - 1]$.
    $factor = range(\mathbf{c}[i : i + n - 1])$.
    $s[i]$=LCS($\mathbf{c}[i : i + n - 1]$, $\mathbf{v}$) $\times$ $factor$.
end for
return  Periods corresponding to top $k$ values in $s$.
```

Algorithm 1 describes bursty pattern detection in a single time series. For each block **b** on a fitted curve **c**, we first normalize it. Then the modified LCS procedure described above finds out the number of matches between the template and the normalized block **b**. By this step, we can find out, in the time series, the locations corresponding to bursty patterns. Taking the degree of burst into account, the number of matches in each block is multiplied by the range of values in that block (the one before normalization), such that greater bursts will be ranked higher.

Algorithm 2 Correlated Abnormal Patterns Detection in Multidimensional Time Series (**CAPD-MDTS**)

```
Input:Multidimensional curves $C$
Output: Periods when correlated anomalies appear
for each dimension $\mathbf{c}_i$ do
    Time points of burst $L_i$=BD-STS($\mathbf{c}_i$)
end for
$n$ = length of $C$, $w$ = time frame length (set to 5)
$S = \emptyset$    //   set of periods to return
for $b = 1 \rightarrow n - w + 1$ do
    $S = S \cup \{[b, b + w - 1]\}$ if $|\{x \in L_i : i = 1, 2, 3, x \in [b, b + w - 1]\}| == 3$
end for
return  $S$
```

After we obtain a list time points corresponding to the top k bursts in each of the dimensions, we need to find out the time windows corresponding to joint increases in all three dimensions. By the first step, we know in each dimension the time when the bursty patterns appear, along with their intensities of burst. In the experiments, we take the top 5 time points. Then we slide a window of a certain size over the time axis. At each point, we find out how many top ranked locations in all dimensions are in the time frame specified by the current time window. A time window is reported if all three dimensions have bursty patterns falling into the window. These steps are formally described in Algorithm 2.

A running example based on the review data is shown in Figure 2. The length of the time window in time series construction is chosen to be 60 days. This example is also discussed in more detail in the experiment section. Each dimension of the time series is plotted in dark points (upper box - rating, middle box - number of reviews, lower box - ratio of singleton review). The solid lines are the fitted curves (to be discussed in the next subsection). We use red vertical dash lines to highlight one of the suspicious blocks detected in time series by the proposed approach. The significant joint bursty pattern locates in $\{19 \rightarrow 24\}$ (from Oct 13, 2005 to Sep 12, 2006), as enclosed by the pair of vertical lines. The three curves all go up in this interval.

3.2.3 A Hierarchical Framework for Robust Singleton Review Spam Detection

Given the review records of a store, one can construct multiple time series using different time window sizes (resolutions). If the window size is set too small, the general trend of a time series would be buried in a large number of fluctuations, which might cause high false positive rate. Therefore, we propose a hierarchical framework, which incorporates Algorithm 2 to robustly detect SR spam attacks. We summarize this hierarchical SR spam detection algorithm in Algorithm 3. We first smooth out short-term fluctuations using a larger window (lower resolution). Then we fit curves using these time series and use Algorithm 2 (**CAPD-MDTS**) to detect any suspicious periods with correlated abnormal patterns, which indicate the high likelihood of SR spam attacks. A smaller window size (higher resolution) can be used to reveal more details (e.g. the exact time of the burst). This is accomplished by constructing new time series with a higher resolution on the detected periods, and detecting any finer suspicious period. This process continues until one reaches the desired resolution such that the time of SR spam attacks can be easily pinpointed.

Algorithm 3 Multi-Scale Spam Detection Algorithm

```
Input: Reviews data of a store, initial window size $\Delta t$,
    time span $I$ when all reviews are collected.
Output: Detected time intervals of spam activities.
Initialize time interval set $S_0 = \{I\}$. Scale $\ell = 0$.
while $\Delta t$ not small enough do
    $\ell = \ell + 1$, $S_\ell = \emptyset$.
    for Each time interval $I \in S_{\ell-1}$ do
        Construct time series $F(I, \Delta t)$.
        Fit a curve for each dimension of $F(I, \Delta t)$.
        Sample the curves to obtain clean time series $C$.
        $S_\ell = S_\ell \cup$ CAPD-MDTS($C$).
    end for
    Decrease window size $\Delta t$,
end while
return  $S_\ell$
```

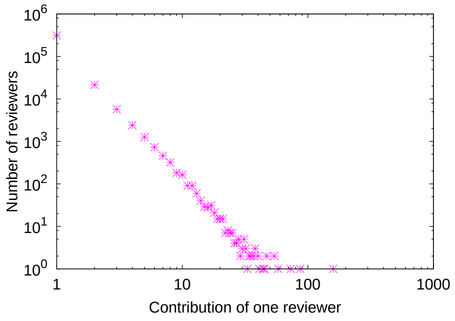


Figure 3: Contributions of reviewers

4. EXPERIMENTS

In this section, we first describe the dataset we use, then we give a couple of case studies to provide evidences of spamming activities caught by the proposed method.

4.1 Review Data Description

The review data we use in the experiments is a snapshot of a review website (`www.resellerratings.com`) on Oct 6th, 2010 [1]. It contains 408,469 reviews written by 343,629 reviewers (identified by their IDs on the website) for 25,034 stores. 310,499 reviewers (> 90%) wrote only one reviews and about 76% (310,499/408,469) of the reviews are SRs. The distribution of the number reviewers writing a number of reviews is plotted in logarithm scale in Figure 3. As we can see, the relation between these two quantities roughly follows the power distribution. This is also observed in [9]. The main body of the data consists of reviews, along with information about stores and reviewers. For each review we keep the following information: its rating (ranging from 1 to 5), the posting date and whether it is an SR.

4.2 Human Evaluation

In this section, we report the experimental results of human evaluation of the detected suspicious stores and reviews. We employ three human evaluators in this experiment.

4.2.1 Suspicious Store Detection

One way to use the algorithm is to run it against the reviews for a store to detect any singleton spam attack. We focus on stores with large number of SRs, so in the evaluation we select top 53 stores, each of which has more than 1,000 reviews. We ask human evaluators to read the reviews from all 53 stores and make decisions regarding the suspiciousness of these stores. If two or more evaluators vote a store as being likely to have committed an SR spam attack, we tag it to be a likely dishonest store. According to the human evaluation, there are a total of 29 stores having at least two votes. Out of the 53 stores, the proposed algorithm labels 36 ones as suspicious stores and the rest as normal ones. Out of the 36 detected ones, 22 stores have at least two votes for being suspicious. The proposed algorithm misses 7 suspicious ones. The recall is 75.86% (22/29), indicating that the proposed algorithm can catch most of the stores involved in SR spam attack. The precision is 61.11% (22/36). Though this precision looks a bit low, since our goal is to identify suspicious stores for human experts to investigate further, the proposed approach only enlarges the suspicious set moderately with a decent recall.

Table 1: Human evaluation results on stores

	Evaluator 1	Evaluator 2	Evaluator 3
Evaluator 1	17	14	16
Evaluator 2	-	20	19
Evaluator 3	-	-	24

Table 2: Human evaluation results on reviews

	Evaluator 1	Evaluator 2	Evaluator 3
Evaluator 1	59	20	28
Evaluator 2	-	41	38
Evaluator 3	-	-	72

Table 1 shows the agreement between evaluators when evaluating the detected stores. The numbers on the diagonal show how many stores each evaluator considers as dishonest. For example, evaluator 1 regards 17 out of 53 stores as suspicious ones. The off-diagonal numbers give how many stores that both evaluators in that row and column identify as dishonest stores. For example, the number on the intersection of Evaluator 1 and Evaluator 2 means that both evaluators 1 and 2 agree upon 14 stores that are suspicious stores. In any case, there are 26 stores at least one of the evaluators regarding it to be suspicious. Comparing the off-diagonal numbers with the diagonal numbers shows the limitation of the content-based approaches. Even human evaluators examining the contents cannot reach agreement a lot of the times as these cases are often very subtle.

4.2.2 Singleton Reviews on a Detected Store

We also ask three human evaluators to examine 147 reviews contained in the detected time window of burst given in the first case study (see next section). Each review is given a score (0-negative, 0.5-possibly, 1-positive) indicating the degree of being regarded as a spam review by each evaluator. Lastly, for each review, the scores from three evaluators are added up to get the final score. Among the 147 reviews, 43 reviews (38 are SR) have final score at least 2, and 12 reviews (11 are SR) have final score equal to 3. This indicates that lots of spam identified are indeed SR and the proposed algorithm can locate the period when SR are more likely to happen.

On the other hand, we can also use similar evaluation technique as in Table 1. Table 2 shows the results of human evaluation on spam reviews. Evaluator 1 tags 59 out of 147 reviews as spam reviews, while the other two regard 41 and 72 reviews as spam reviews, respectively. There are 98 reviews that at least one of the evaluators regard as spam. Similarly, the numbers off the diagonal show the agreement between evaluators. Again, this table shows that it is not easy for human beings to reach agreements on whether a review is an SR spam, and content-based methods will be less effective in the detection of this kind of spams.

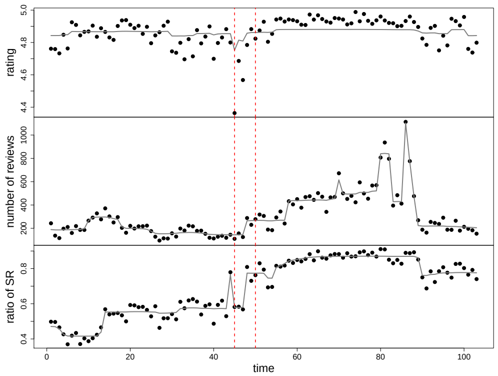
4.3 Spam Detection Case Study

In this section, we closely study the evidences of SR attacks committed by several stores.

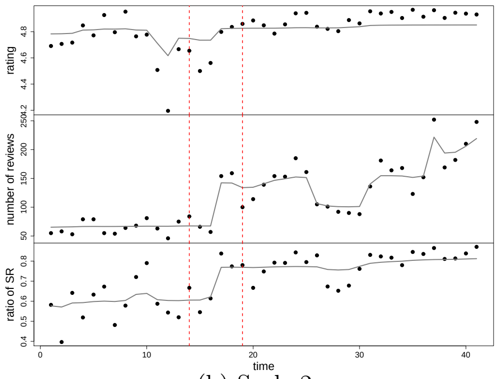
4.3.1 First Case Study

The results of running the proposed multidimensional multiscale detection algorithm on the reviews of a store (id=24811)

[1] Thanks to Keith Nowicki



(a) Scale 1



(b) Scale 2

Figure 4: Anomaly detection on multi-scale multi-dimensional time series

are shown in Figure 4. The multidimensional time series in the first subfigure (Figure 4(a)) is produced using a larger time window (30 days) with review data from Apr 2002 and to Aug 2010. The format of this figure is the same as that of Figure 2. In this higher level of detection, we find a sharp increase in {45 → 50}, which corresponds to the time interval from Oct 30, 2005 to Mar 30, 2006. Notice the burst occurs in a two-month period. We construct another 3-dimensional time series from the review data in the detected time window to find more details of the burst. With the window size set to 15 days, we run the detection algorithm again. The block {14 → 19} (from Dec 17, 2005 to Mar 3, 2006) with suspicious activities are found and highlighted in Figure 4(b). Note that this detected time window is smaller than the previous one. In this time interval, the number of singleton reviews increases from 57 to 154, the rating goes up from 4.56 to 4.79, and the ratio of singleton reviews goes up from 61% to 83%. These all happen in a two-weeks period. It looks like the ratings were bolstered by the sudden increase of singleton reviews.

However, one might still not be convinced that there are probably spams activities in the detected time intervals. We provide further evidences by analyzing the review contents. Note that looking for these evidences is *not* part of the proposed algorithm, which only uses the reviewers' behaviors for detection. This is only for the purpose of validation. We find out that around the time when the bursts in all three dimensions are detected, the phrases "customer service" and

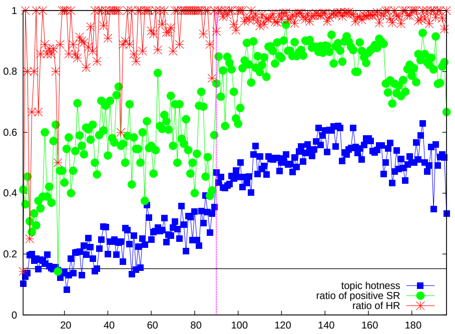


Figure 5: Topic Hotness Trend

"customer support" are unusually frequent among the SRs. Such correlation indicates that there could be spammers giving undeserving high ratings to "customer service" of that store. Next we study this correlation quantatively.

When constructing multidimensional time series, we divide the time from Apr 2002 to Aug 2010 into intervals of two weeks. For each interval, we calculate the "hotness" of the topic "customer service". The "hotness" of a topic is the ratio of reviews about that topic to all reviews in a certain period. If a review contains one of the phrases "customer service" and "customer support", we consider it to be related to that topic. In Figure 5, we show the trend of the hotness of the topic in the blue curve with solid squares. One can see that there is a burst of topic hotness occurs at time 90 (Feb 06, 2006, indicated by the dashed line). Note that this burst occurs in a two-week long period with the hotness goes up from 35% to 46%. Also, note that the time of this burst coincides with that of the burst detected in the multidimensional time series by the proposed algorithm. This makes the detected time interval look suspicious. The black horizontal solid line shows the topic hotness calculated from all reviews except those from the store being investigated. We can see that, on average, less than 16% of the reviews mentions the phrases. This number is calculated using 376,758 reviews out of the total 408,470 reviews, so it well represents reviewers' general interests of this topic. By comparison, we can see that the hotness within this store is twice as high as the average level. This is unlikely in normal business, since it is quite hard to gain the recognition of "customer service" from real customers in two weeks. After that time, the topic hotness keeps going up and is far higher than the average level. In particular, one out of two reviews is talking about "customer service" on average.

Besides "topic hotness", we consider two other reviewer behaviors. The green curve with solid circles shows the ratio of singleton 5 star reviews to the topic-related reviews. We can see that from the time Feb 06, 2006 on, this ratio is rather high, namely, more than 80% of the singleton reviews are related to "customer service". We can conclude that the burst and hotness of the topic is supported by the burst and high volume of singleton reviews. Lastly, the red curve with stars gives the ratio of reviews which are written by "hurry reviewers" (HR) to the singleton 5 star reviews. We define an HR to be a reviewer who writes a review on the same day she registers her id. From the figure, one can tell that, from Feb 06, 2006 on, a high percentage (over 90%) of 5 star singleton reviews about "customer service" are produced by

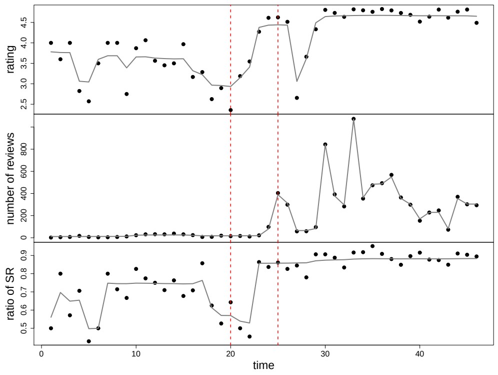


Figure 6: Bursty Patterns Detected in Store 24938

"hurry reviewers". Since at least for those who registered in 2005 never write another review in the following 5 years, this is quite dubious. As a way of validation, we read reviews of the store in the period of topic hotness burst. We found a reviewer once disclosed the fact that the store emailed her for a favorable rating. The reviewer had an unpleasant experience with that store and got customer service only after she low-rated it on the review website.

4.3.2 Second Case Study

When we try to investigate a store *meritline* with high SR spams identified by the proposed algorithm, we find out it also operates under another name *cdrdvdrmedia*. Hence this case of spamming is quite interesting. The following facts support this observation: first, the addresses of the two companies are the same[2][3]. Second, on the review website we are studying, *meritline* is an alien of *cdrdvdrmedia*. Third, according to a domain analysis website, these two stores have the same Google analytics account[4]. Lastly, one reviewer says the package and receipt she received were from *meritline* though she shopped with *cdrdvdrmedia*[5]. We perform the proposed multidimensional times series analysis on the reviews for meritline. Figure 2 (Section 3.2) shows the time interval when an SR spam attack is likely to have happened. *cdrdvdrmedia* sells the same set of products as *meritline* does, but with a much lower rating. There are only 48 reviews in near 8 years (from Aug-2002 to Jan-2010). The average rating of the store is only 3.06 and people are talking about credit card problems, low-quality products and customer service. Therefore, the high volume of reviews and good rating for *meritline* are quite suspicious.

4.3.3 Third Case Study

We find another store (*supermediastore*) which is likely involved in singleton review spamming. The multidimensional time series for the store and the detected bursty patterns are shown in Figure 6. This store is probably owned by the same owner as *meritline* and *cdrdvdrmedia*. This is supported by at least two forum posts[6]. We also find an interesting review[7] telling that the reviewer was cheated by *supermediastore* when it tried to entice her into spamming. The reviewer once received an email from the store about writing a review for it. In return, the reviewer would receive a "gift", which she never receive. This is a direct evidence that this store is hiring/enticing people to write favorable reviews. This review is written during the time when there is a burst of singleton reviews.

5. RELATED WORK

There is few existing algorithm specifically designed for singleton review spams detection. In [11], the authors consider singleton review spams in their data collecting process. They "artificially" construct singleton review spams for evaluation, while in this paper, the singleton review spams occur in a real-world dataset. Therefore, the singleton reviews they construct don't exhibit the temporal features that real-world singleton review spams should have. In [14], the authors attempt to detect hotels which are more likely to be involved in spamming. They construct several features, based on which two ranking algorithms detect the most suspicious hotels. Their method is not comparable with the one we proposed, because their method is supervised (they need to know if certain positive reviews are shill reviews reacting to negative reviews). As their method provides a ranking list of hotels, one will not be able to tell how many hotels need further inspection, while the proposed method gives a list of suspicious stores for further human evaluation.

Another work considers reviewers' behaviors by introducing a social graph connecting reviewers, their reviews and stores [5]. They discover the reinforcement relations of reviewers' trustiness, reviews' honesty, and stores' reliability. They use such relations to discover suspicious spammers. However, their method is unable to handle singleton reviews, because SRs have insufficient information in the constructed graph to infer their trustiness. Another work [4] uses reviewers' behaviors as indicators of spamming. The features include multiple similar rating on a single product, similarity between reviews written by a single reviewer, etc. One will not be able to compute these features for SRs.

Another work [1] studies group spammers. They propose a three-step method to detect group of spammers. They first use a frequent pattern mining algorithm to find out groups of reviewers who frequently write reviews together. Then they construct features to find out the most likely group spammers. This method is novel in detecting a new form of spam activity. Yet it still does not address the singleton review spam problem, because only group of reviewers write reviews together at least 3 times will be considered suspicious. Similarly, the rule-based algorithm for unexpected review activities detection proposed in [10] would also fail, because during the rule discovery step, SRs will be pruned due to the support threshold.

Earlier works on spam detection mainly rely on review similarities to construct features for spam reviews. For example, the work in [9] categorizes spam reviews into three

[2] www.cdrdvdrmedia.com/contact-us.html

[3] www.la.bbb.org/business-reviews/General-Merchandise-Retail-By-Internet/Meritline-in-City-of-Industry-CA-13135057

[4] domaintraker.com/meritline.com

[5] www.resellerratings.com/store/view/CDRDVDRMEDIA_17/page/1, see username "sableman"

[6] forum.doom9.org/archive/index.php/t-36023.html and forum.videohelp.com/threads/143262-Meritline-Very-Disappointed

[7] www.resellerratings.com/store/view/Supermediastore/page/895, see the review from the ID "defile"/

types: untruthful opinions; reviews on brands only; non-reviews. To detect these different types of spams, they use simple duplicate detection, feature construction and logistic regression. There are some drawbacks. First, it needs to collect a considerable amount of spam reviews manually. This is time consuming, even if it only looks at those reviews which are obviously spams. Second, it only deals with a small portion of all the reviews (neither the duplicated nor the manually labeled reviews represent a large enough fraction of the reviews). Finally, it does not address the singleton reviews spam problem, for example, the reviewers' centric features again rely heavily on the basis that a reviewer posts more than one review.

There are researches on multidimensional or multivariate time series anomaly detection [15, 6, 2]. However, we cannot directly apply these methods to the singleton spam detection problem. The methods in [2, 6] are designed for general abnormal pattern mining in multidimension time series, and [15] proposes to search all subspaces of all attributes in a relation (in the database sense), where the subspaces reported contain anomaly. We focus on detecting correlated abnormal patterns in all three dimensions, while a general detection algorithm will produce many abnormal patterns irrelevant to the problem in this paper, and therefore increase false positive rate.

Regarding burst detection, there are some excellent related works [7, 13, 3]. The work in this paper is not directly comparable to any general burst detection or curve fitting algorithm for the following reasons: first, the algorithm we propose is a general framework, which can incorporate any burst detection algorithm. Second, the problem we study in this paper is not merely a burst detection problem in time series, since we are not given the time series. Instead, we need to model the behaviors of reviewers and construct the time series. Also we need to identify the way in which abnormal patterns are uniquely correlated.

6. CONCLUSION

This paper studies the problem of singleton review spam detection, which is both difficult and important to solve. We transform this problem to a temporal pattern discovery problem. We identify three aggregate statistics which are indicative of this type of spam attack, then we construct a multidimensional time series using these statistics. We design a multi-scale anomaly detection algorithm on multidimensional time series based on curve fitting. Experimental results show that the proposed algorithm is effective in detecting singleton review spams.

7. ACKNOWLEDGEMENTS

This work is supported in part by NSF through grants IIS-0905215, IIS-0914934, CNS-1115234, DBI-0960443 and OISE-1129076, US Department of Army through grant W911NF-12-1-0066, and Google Mobile 2014 Program.

8. REFERENCES

[1] Mukherjee A, Liu B, Wang J, Glance N, and Jindal N. Detecting group review spam. WWW '11.

[2] Vahdatpour A and Sarrafzadeh M. Unsupervised discovery of abnormal activity occurrences in multi-dimensional time series, with applications in wearable systems. In *SDM'10*.

[3] Chandra E and John W. E. bcp: An R Package for Performing a Bayesian Analysis of Change Point Problems. *Journal of Statistical Software*, 2007.

[4] Lim E-P, Nguyen V-A, Jindal N, Liu B, and Lauw H W. Detecting product review spammers using rating behaviors. CIKM '10.

[5] Wang G, Xie S, Liu B, and S.Yu P. Identify online store review spammers via social review graph. In *ICDM'11*.

[6] Cheng H, Tan P, Potter C, and A. Klooster S. Detection and characterization of anomalies in multivariate time series. In *SDM'09*.

[7] Jon K. Bursty and hierarchical structure in streams. KDD '02.

[8] S. Karlin and H. M. Taylor. *A First Course in Stochastic Processes*. Academic Press, 2 edition, 1975.

[9] Jindal N and Liu B. Opinion spam and analysis. WSDM '08.

[10] Jindal N, Liu B, and Lim E-P. Finding unusual review patterns using unexpected rules. CIKM '10.

[11] M Ott, Y Choi, C Cardie, and J. T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. HLT '11.

[12] Michail V, Marios H, Dimitrios G, and Eamonn K. Indexing multi-dimensional time-series with support for multiple distance measures. KDD '03.

[13] Michail V, Christopher M, Zografoula V, and Dimitrios G. Identifying similarities, periodicities and bursts for online search queries. SIGMOD '04.

[14] G Wu, D Greene, and P Cunningham. Merging multiple criteria to identify suspicious reviews. RecSys '10.

[15] Li X and Han J. Mining approximate top-k subspace anomalies in multi-dimensional time-series data. In *VLDB'07*.

View publication stats