Exploiting Burstiness in Reviews for Review Spammer Detection

Geli Fei[1] Arjun Mukherjee[1] Bing Liu[1] Meichun Hsu[2] Malu Castellanos[2] Riddhiman Ghosh[2]

[1]Department of Computer Science, University of Illinois at Chicago, Chicago, USA
[2]HP Labs, Palo Alto, California, USA
gfei2@uic.edu, arjun4787@gmail.com, liub@cs.uic.edu
{meichun.hsu, malu.castellanos, riddhiman.ghosh}@hp.com

Abstract

Online product reviews have become an important source of user opinions. Due to profit or fame, imposters have been writing deceptive or fake reviews to promote and/or to demote some target products or services. Such imposters are called review spammers. In the past few years, several approaches have been proposed to deal with the problem. In this work, we take a different approach, which exploits the burstiness nature of reviews to identify review spammers. Bursts of reviews can be either due to sudden popularity of products or spam attacks. Reviewers and reviews appearing in a burst are often related in the sense that spammers tend to work with other spammers and genuine reviewers tend to appear together with other genuine reviewers. This paves the way for us to build a network of reviewers appearing in different bursts. We then model reviewers and their co-occurrence in bursts as a Markov Random Field (MRF), and employ the Loopy Belief Propagation (LBP) method to infer whether a reviewer is a spammer or not in the graph. We also propose several features and employ feature induced message passing in the LBP framework for network inference. We further propose a novel evaluation method to evaluate the detected spammers automatically using supervised classification of their reviews. Additionally, we employ domain experts to perform a human evaluation of the identified spammers and non-spammers. Both the classification result and human evaluation result show that the proposed method outperforms strong baselines, which demonstrate the effectiveness of the method.

Introduction

There is a growing trend that people rely on online product reviews to make purchase decisions. Products with a large percentage of positive reviews tend to attract more customers than products without a large percentage of positive reviews. Due to the reason of profit or fame, imposters have tried to cheat the online review system by writing fake or deceptive reviews to deliberately mislead potential customers. They may give unfair positive reviews to some products in order to promote them and/or give malicious negative reviews to some other products in order to damage their reputations. These imposters are called review or opinion spammers (Jindal and Liu 2008).

In the normal situation, reviews for a product arrive randomly. However, there are also areas (time periods) where the reviews for a product are bursty, meaning that there are sudden concentrations of reviews in these areas or time periods. We call such areas *review bursts*. A review burst can either be due to a sudden increase of popularity of a product or because the product is under a spam attack. For example, a product may suddenly get popular because of a successful TV commercial. Then, a large number of customers may purchase the product and write reviews for the product in a short period of time. Most reviewers in this kind of bursts are likely to be non-spammers. In contrast, when a product is under spam attack, a number of spam or fake reviews may be posted (posting a single review may not significantly affect the overall sentiment on the product). These two possibilities lead to an important hypothesis about review bursts, i.e., reviews in the same burst tend to have the same nature, meaning that they are either mostly from spammers or mostly from genuine reviewers. In this paper, we exploit review bursts to find spammers who wrote fake reviews in bursts.

In the past few years, researchers have designed several methods for detecting review spam or review spammers. Most existing works focused on analyzing one review or one reviewer at a time, neglecting the potential relationships among multiple reviews or reviewers (Jindal and Liu 2008; Lim et al. 2010; Jindal, Liu, and Lim 2010; Li et al. 2011; Ott et al. 2011). Although (Wang et al. 2011) studied the problem of detecting online store review spammers by considering the relationships of reviewers, reviews, and stores, they do not consider the relationships among reviewers (or reviews) themselves. Our proposed method considers such relationships by linking reviewers in a burst. Furthermore, their method only produces a ranking of reviewers based on their computed spam scores, but our proposed method assigns a spam or non-spam label to each reviewer.

To exploit the relatedness of reviews in bursts, we propose a graph representation of reviewers and their relationships, and a graph propagation method to identify review spammers. Several spamming behavior indicators are also proposed to help the propagation algorithm.

In summary, this research makes the following main contributions:

(1) It proposes an algorithm to detect bursts of reviews using Kernel Density Estimation and also several features as indicators for use in detecting review spammers in review bursts.

(2) It proposes a data model based on Markov Random Fields, and employs feature induced message passing in the loopy belief propagation framework to detect review spammers. Although (Wang et al. 2011) also used a graph to link reviewers, reviews and stores for detecting store spammers, as we discussed above, their method does not identify spammers but only rank them.

(3) It proposes a novel evaluation method to evaluate the detected spammers automatically using supervised classification of their reviews. Since the proposed method is like clustering, we can build a classifier based on the resulting clusters, where each cluster is regarded as a class. The key characteristic of the approach is that the features used in detecting spammers are entirely different from the features used in classification (i.e., there is no feature overlap). This approach is objective as it involves no manual action.

To the best of our knowledge, the proposed method has not been used before. For evaluation, we use Amazon reviews. Our classification based method shows high accuracy, which gives us good confidence that the proposed graph propagation method is working. The strong results are further confirmed by human evaluation.

Related Work

The problem of detecting deceptive or fake reviews (also called *opinion spam*) was proposed in (Jindal and Liu 2008). The existing approaches can be categorized into two main types: supervised methods and unsupervised methods. The approach in (Jindal and Liu 2008) is based on supervised learning. It builds a classifier using certain types of duplicate and near-duplicate reviews as positive training data (fake reviews) and the rest as the negative training data (non-fake reviews). Ott et al. (2011) employed standard word and part-of-speech (POS) n-gram features for supervised learning using crowdsourced fake reviews obtained from Amazon Mechanical Turk and some selected reviews from Tripadvisor.com as non-fake reviews. Li et al. (2011) also used supervised learning. In their case, the training and testing reviews are labeled manually. Mukherjee et al. (2013) classified Yelp filtered and unfiltered reviews, and performed a comparative study of commercial vs. crowdsourced fake reviews for supervised classification. These approaches all assume there are reliably labeled reviews.

In the unsupervised approach, Jindal, Liu, and Lim (2010) proposed a method based on mining unexpected rules. Lim et al. (2010) studied spammer detection by using some predefined types of behavior abnormalities of reviewers. Wang et al. (2011) used a graph-based method to find fake store reviewers by considering the relationship among reviewers, reviews and stores. As stated in their paper, due to the difference between store reviews and product reviews, their methods are specific for store review spammers. Xie et al. (2012) studied the detection of a special group of review spammers who only write one review which they call singleton review spam. Since the authors only deal with reviewers with one review, our research can be seen as complementary to their work. In (Mukherjee, Liu, and Glance 2012), the authors studied the problem of detecting fake reviewer groups by considering both group and individual reviewer behavioral features. Feng et al. (2012) first studied the distributional anomaly of review ratings on Amazon and TripAdvisor, and then proposed strategies guided by the statistics that are suggestive of the distributional anomaly to detect spam reviews.

Burst Detection

In this section, we introduce the method for burst detection using Kernel Density Estimation (KDE) techniques. KDE is closely related to histograms, but can be endowed with properties such as smoothness and continuity, which are desirable properties for review burst detection in a product.

Kernel Density Estimation

Given a sample $S=\{x_i\}_{i=1\ldots N}$ from a distribution with density function $f(x)$, an estimate $\hat{f}(x)$ of the density at x can be calculated using

$$\hat{f}(x) = \frac{1}{N}\sum_{i=1}^{N} K_h(x - x_i)$$

where K_h is called the *scaled kernel* (sometimes called the "window" function) with a bandwidth (scale) h such that $K_h(t) = hK(t/h)$. K is called the *kernel*, which should satisfy $K(u) \geq 0$ and $\int K(u)du = 1$. We can think of the above equation as estimating the pdf by averaging the effect of a set of kernel functions centered at each data point. Kernel density estimators asymptotically can converge to any density function with sufficient samples as reported in (Scott 1992; Duda, Stork and Hart 2000). This property makes the technique quite general for estimating the density of any distribution.

Burst Detection Method

Given a product p which has a set of m reviews $\{p_1, \ldots, p_m\}$, and each review has a review date associated with it $\{t_1, \ldots, t_m\}$. So the duration of the product dur is computed by $t_m - t_1$, which is considered as the difference between the latest review date and the first review date.

We first divide the life span of the product into k small sub-intervals or bins by choosing a proper bin size BSIZE. In this paper, we set BSIZE equal to two weeks. Then we compute the average number of reviews within each bin with $avg_{rev} = m/k$.

For each bin i, let $H_i = \{p_j | t_j \in (a_{i-1}, a_i], i \in \{1, \ldots, k\}\}$ be the set of reviews that fall into this bin, where $a_i = i * BSIZE$.

We then normalize the duration of the product to [0, 1] by dividing each interval by dur such that $a_i = a_i/dur$.

We use the Gaussian kernel in our method and thus $x_1 = a_1, \ldots x_k = a_k$ can serve as the binned samples over the range [0, 1] with weights $w_1 = |H_1|, \ldots, w_k = |H_k|$. The estimate is given by:

$$KDE(x) = \hat{f}_h(x) = \frac{1}{h\sum_{i=1}^{k} w_i}\sum_{i=1}^{k} w_i K\left(\frac{x - x_i}{h}\right)$$

where $K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$, h is the bandwidth, which controls the smoothness of the estimate. We set the bandwidth experimentally by trying different values and chose the one which made the estimation not too jagged or too smooth.

By taking the derivative of the density function and setting it to zero, we find a set of peak points $\{x_{p1}, \ldots, x_{pt}\}$, with each peak point x_{pj} falling into some bin i.

Since our objective is to detect bursts, which are the periods of time a product sees sudden increases in the number of reviews, so we first remove those peak points that fall in bins with $|H_i| \leq avg_{rev}$. Also, there are cases that some areas only contain one review. We get peak points for these areas and discard them as we do not consider them as representing real bursts. Then for each of the remained peak points, we keep including its left bins and right bins e as long as $|H_e| \neq 1$ and $|H_e| > avg_{rev}$, and thus all reviews within these bins form one burst of reviews that we are interested in.

Spammer Behavior Features

In this section, we present the spammer behavior features or indicators that we use in this work. All the feature values that we compute are normalized to [0, 1]. Note that our current features do not apply to reviewers who wrote only one review as there is little behavior embedded in a single review. There is an existing method that deals with such reviewers (Xie et al. 2012). Proposing a generic framework to deal with both kinds of reviewers will be a part of our future work. Below, we list our features.

Ratio of Amazon Verified Purchase (RAVP): When a product review is marked "Amazon Verified Purchase", it means that the reviewer who wrote the review has purchased the item at Amazon.com. So we can expect that a genuine reviewer should have higher ratio of "Amazon Verified Purchase" reviews than spammers as spammers usually do not buy the products that they review. RAVP is computed using one minus the number of "Amazon Verified Purchase" reviews that a reviewer wrote divided by the total number of reviews that he/she wrote.

$$RAVP(a) = 1 - \frac{verified(V_{a*})}{|V_{a*}|}$$

where V_{a*} represents the set of all reviews that reviewer a wrote towards all products, and $verified(V_{a*})$ represents the number of AVPs among V_{a*}. We use $|*|$ to indicate the number of elements within a set. Note that if a review is not marked Amazon Verified Purchase, it doesn't mean that the reviewer has no experience with the product – it only means that Amazon.com couldn't verify that.

Rating Deviation (RD): A reasonable reviewer is expected to give ratings similar to other reviewers of the same product. As spammers attempt to promote or demote products, their ratings can be quite different from other reviewers. Rating deviation is thus a possible behavior demonstrated by a spammer. We define the rating deviation of reviewer a as follows:

$$RD(a) = \underset{p \in P_a}{avg} \frac{|r_{ap} - \bar{r}_{ap}|}{4}$$

where r_{ap} refers to the rating given by reviewer a towards product $p \in P_a$, which is the set of products that he/she has reviewed, and $\bar{r}_{ap}$ refers to the average rating of the product given by other reviewers than a. We normalized the value by 4, which is the maximal possible rating deviation on a 5-star rating scale. Finally, we compute the average deviation of all reviewer a's reviews.

Burst Review Ratio (BRR): This feature computes the ratio of a reviewer's reviews in bursts to the total number of reviews that he/she wrote. Since we expect the arrival of normal reviews to be random, if a reviewer has a high proportion of reviews in bursts, he/she is more likely to a spammer. BRR of reviewer a is computed as follows:

$$BRR(a) = \frac{|B_{a*}|}{|V_{a*}|}$$

where B_{a*} represents the set of reviews that reviewer a wrote that have appeared in review bursts.

Review Content Similarity (RCS): Review content similarity measures the average pairwise similarity of all reviews that a reviewer wrote. Since spammers normally do not spend as much time as genuine reviewers in writing a completely new review, the words they choose every time are expected to be similar. We use the bag-of-words model to represent each review text and the cosine similarity between two reviews as their content similarity. So RCS of a reviewer a is computed as shown below:

$$RCS(a) = \underset{v_{a,i}, v_{a,j} \in V_a, i<j}{avg} \text{cosine}\left(v_{a,i}, v_{a,j}\right)$$

where V_a is the set of reviews that reviewer a wrote.

Reviewer Burstiness (RB): If the reviews appearing in some product review bursts happen to be the reviews in a reviewer's own review burst, he/she is more likely to be a spammer. We thus use reviewer burstiness to measure this behavior, and RB is computed as follows:

$$RB(a) = \begin{cases} 0 & L(B_{a*}) - F(B_{a*}) > \lambda \\ 1 - \dfrac{L(B_{a*}) - F(B_{a*})}{\lambda} & otherwise \end{cases}$$

where $L(B_{a*})$ and $F(B_{a*})$ are the latest and earliest time of the reviews that reviewer a wrote that appears in the burst respectively. λ is the time window parameter representing a burst in a customer's own review pattern. In this paper, we set λ equal to two months based on the observation in (Xie et al. 2012).

In what follows, we will use the ratio of Amazon Verified Purchase (RAVP) as the state prior because we believe that it is a stronger and reliable feature than the other four. Moreover, we use the expected value of all the other four features for a reviewer a as an overall spamming indicator (OSI) of the reviewer's spamming behavior.

$$OSI(a) = \frac{RD(a) + BRR(a) + RCS(a) + RB(a)}{4}$$

Burst Review Spammer Detection Model

In this section, we present the models that we employ to model the identity (spammer or non-spammer) of each reviewer and the networks that reviewers create within bursts.

We begin by describing the Markov Random Field (MRF) model, which is a set of random variables having a Markov property described by an undirected graph. We will use a MRF to model the identity of reviewers in the graphical form. Then we describe two versions of the Loopy Belief Propagation algorithm, and show how the algorithm could be applied on a MRF and used to detect review spammers in our problem.

The Markov Random Field Model

Markov random fields (MRFs) are a class of probabilistic graphical models that are particularly suited for solving inference problems with uncertainty in observed data. They are widely used in image processing and computer vision, e.g., image restoration and image completion.

A MRF comprises two kinds of nodes – hidden nodes and observed nodes. Observed nodes correspond to the values that are actually observed in the data. For each observed node, there is a hidden node which represents the true state underlying the observed value. The state of a hidden node depends on the value of its corresponding observed node as well as the states of its neighboring hidden nodes. These dependencies are captured via an edge compatibility function $\varphi(\tau, \tau')$. $\varphi(\tau, \tau')$ gives the probability of a hidden node being in state τ given it has a neighboring hidden node in state τ'. $\phi(\tau, \omega)$ gives the probability of a node being in state τ given its corresponding observed node is ω. With each hidden node i, we also associate a belief vector $\boldsymbol{b}_i$, such that $\boldsymbol{b}_i(\tau)$ equals the probability of node i being in state τ (which we call the belief of node i in state τ).

In this paper, we model the reviewers in bursts and their co-occurrences as a MRF. By co-occurrence we mean that some reviewers who wrote reviews in the same burst. We create a hidden node for each reviewer to represent his/her real yet unknown identity, which can be in any of three states – non-spammer, mixed and spammer. The reason we use mixed is due to the fact that some reviewers sometimes write fake reviews for profit and other times are legitimate buyers and write genuine reviews. The co-occurrence between two reviewers within the same burst is represented by an edge connecting their corresponding hidden nodes, so all reviewers that appear in the same burst form a clique. Here we do not distinguish how many times two reviewers appear in the same bursts. Also, as mentioned above, each hidden node is also associated with an observed node, which corresponds to our observation of its state in the data.

To completely define MRF, we need to instantiate the propagation matrix. An entry in the propagation matrix $\varphi(\tau, \tau')$ gives the likelihood of a node being in state τ given it has a neighbor in state τ'. A sample instantiation of the propagation matrix is shown in Table 1.

Table 1: An example propagation matrix

	spammer	non-spammer	mixed
spammer	0.4	0.25	0.35
non-spammer	0.25	0.4	0.35
mixed	1/3	1/3	1/3

This instantiation is based on the following intuition: In a review burst, a spammer is most likely to work with other spammers in order to create a major impact on the sentiment of the product being reviewed. Due to the fact that reviewers with mixed identity could also act as spammers, a spammer is more likely to appear together with them than genuine reviewers. Likewise, genuine reviewers are most likely to appear together with other genuine reviewers due to the possibility that the product gets popular suddenly; and they are also more likely to appear together with reviewers with mixed identity than with heavy spammers. However, a reviewer with mixed behavior is equally likely to appear with spammers, mixed, or non-spammers.

The Loopy Belief Propagation Algorithm

Loopy belief propagation (LBP) is a message passing algorithm for solving approximate inference problems on general undirected graphs that involve cycles. It is similar to the belief propagation (BP) algorithm that is applied to solve exact inference problems on trees. The LBP algorithm infers the posterior state probabilities of all nodes in the network given the observed states of some of the network nodes.

Now we present how LBP works on detecting spammers in our work. In the algorithm, we introduce message vector $\mathbf{m}_{ij}$, which is a vector of the same dimensionality as the number of states each node can choose from, with each component being proportional to how likely node i thinks that node j will be in the corresponding state. So $\mathbf{m}_{ij}(\tau)$ represents the likelihood that node i thinks node j being in state τ.

LBP with State Prior Only

Pandit et al. (2007) modeled suspicious patterns that online auction fraudsters create as a MRF and employed a LBP algorithm to detect likely networks of fraudsters. In their work, no priors or observed knowledge was used. Each node was initialized to an unbiased state. However, in this paper, we assign a prior state to each hidden node, as fully unsupervised LBP is known to produce poor models (Yedidia, Freeman, and Weiss 2001). We use the ratio of Amazon Verified Purchase (RAVP) as the state prior because we assume that this is a more reliable indicator than other indicators that we designed above. And we use this setting as one of the baselines in our paper.

In (Pandit et al. 2007), the belief at a node i is proportional to the product of all the messages coming into node i:

$$\mathbf{b}_i(\tau) = k \prod_{j \in N(i)} \mathbf{m}_{ji}(\tau)$$

where k is a normalization constant as the beliefs must sum to 1 and $N(i)$ denotes the nodes neighboring i.

The message $\mathbf{m}_{ij}$ from node i to node j can only be sent across the link when all other messages have been received by node i across its other links from neighboring nodes n.

$$\mathbf{m}_{ij}(\tau) = \sum_{\tau'} \psi(\tau, \tau') \prod_{n \in N(i) \backslash j} \mathbf{m}_{ni}(\tau')$$

Note that we take the product over all messages going into node i except for the one coming from node j.

Because there are loops in the graph, this raises the issue of how to initiate the message passing algorithm. To resolve this, we can assume that an initial message given by the unit function (i.e., a node believes any of its neighboring nodes to be in any of the possible states with equal probability) has been passed across every link in each direction, and every node is then in a position to send a message.

LBP with Prior and Local Observation

In this sub-section, we introduce our feature induced message passing strategy in the LBP framework for network inference. Recall that in the MRF framework, $\phi(\tau, \omega)$ gives the probability of a node being in state τ given its corresponding observed node is ω. We use the ratio of Amazon Verified Purchase (RAVP) to initialize ω so that ϕ is considered as a state prior; and in subsequent steps, ω is set to the overall spamming indicator (OSI) of a reviewer, which is considered as the local observation of the state of each node. We believe that such a combination of local observation and belief passing would yield the following benefits: (a) using a strong prior such as RAVP and a local observation OSI will help the belief propagation to converge to a more accurate solution in less time, (b) since we treat OSI as a noisy observation of the real state of its corresponding node, we expect that incorrect inference of the local observation be corrected by considering the relationships between reviewers in such a graph model. After involving the overall spamming indicator (OSI), the belief at a node i is proportional to both the product of the local observation at that node $\phi(\tau,\omega_i)$ and all the messages coming into node i:

$$\mathbf{b}_i(\tau) = k\phi(\tau,\omega_i)\prod_{j\in N(i)}\mathbf{m}_{ji}(\tau)$$

where k is a normalization constant as the beliefs must sum to 1 and $N(i)$ denotes the nodes neighboring i.

$$\mathbf{m}_{ij}(\tau) = \sum_{\tau'}\psi(\tau,\tau')\phi(\tau',\omega_i)\prod_{n\in N(i)\setminus j}\mathbf{m}_{ni}(\tau')$$

Also, due to the cycles in the graph, information can flow many times around the graph. The algorithm is stopped when the beliefs converge (with some threshold), or a maximum limit for the number of iterations is exceeded. Although convergence of LBP is not guaranteed theoretically, in our problem it converges very quickly (within 20 iterations).

State Prior and Local Observation

In this sub-section, we show the method we use to compute $\phi(\tau,\omega)$ given the value of ω, which is either initialized to the ratio of Amazon Verified Purchase (RAVP) or set to a reviewer's overall spamming behavior (OSI). In both cases, ϕ is a real-valued vector of size three. Each component of the vector represents the likelihood of being a spammer, mixed or non-spammer given the value of ω.

Given a normalized value ω, we use a Gaussian distribution $N(\omega,\sigma^2)$ to compute a reviewer's probability of being a spammer, mixed and non-spammer as follows

$$p(x_i|\omega) = k\int_{0.33*i}^{0.33*(i+1)} f(t)dt, (i = 0,1,2)$$

where $f(t)$ is the density function of $N(\omega,\sigma^2)$, and x_i is the random variable representing the possible state of each reviewer, with x_0 representing non-spammer, x_1 representing mixed, and x_2 representing spammer. k is the normalization factor such that the sum of three probabilities equal to one. In our experiments, we pick $\sigma = 0.25$ so that the normal distribution is concentrated around the mean ω.

Experimental Evaluation

We now evaluate the proposed method. We use product reviews from Amazon.com as our experiment data, which were crawled by the authors of (Jindal and Liu 2008). For our study, we used reviews from the software category, which comprises 210,761 reviews, 50,704 reviewers and 112,953 products. After applying the proposed burst detection method, we found 10,251 bursts and 4,465 non-singular reviewers in these bursts. Two types of evaluations are performed: supervised classification and human evaluation. Supervised classification is a new method proposed in this paper.

Evaluation Using Supervised Text Classification

One of the major obstacles towards review spammer detection is the evaluation because there is no ground truth data of spam and non-spam that can be used in model building and model testing. So, researchers have used human evaluation in previous works. However, human evaluation is subjective as different evaluators often have different tolerance levels even if they are given the same set of behavior indicators and reviews of a reviewer.

We thus propose a novel way of evaluating review spammers, which can be considered as complementary to human evaluation, and thus give us more information about whether the detection algorithm is doing a good job or not. First, we assume that if a reviewer is labeled as a spammer, then all his/her reviews are considered as spam reviews and if a reviewer is labeled as a non-spammer, then all his/her reviews are considered as non-spam reviews. Therefore we can treat the spam reviews as belonging to the positive class and non-spam reviews as belonging to the negative class. A classifier can then be built to separate the two classes of reviews. We applied Support Vector Machine (SVM) in the experiments and used the bag-of-words model and unigram Boolean assignment for feature values (TF-IDF based features did poorer). We note that in our detection algorithm, we only used behavior features. However, in the review classification, we use purely linguistic features. If the classification shows good accuracy, we know that the reviews written by reviewers labeled as spammer and non-spammer based on their behaviors are also separable based on their review text. Note that we do not use the mixed class in classification because it contains a mixture of spammer and non-spammers, which are harder to separate.

For the two classes in our model (spammer and non-spammer), we build two classifiers. One is only based on the reviews that have appeared in some bursts. The other is based on all reviews of the spammers and non-spammers regardless whether the reviews appeared in bursts or not. In both cases, we treat the reviews written by spammers as belonging to the positive class and reviews written by non-spammers as belonging to the negative class. The reason for building two classifiers is as follows: Recall in the introduction section, we hypothesized that reviews in each burst are more likely to be of the same nature (spam or non-spam). Those reviews not in bursts are more random because a reviewer may write fake reviews sometimes and also genuine reviews some other times as he/she can be a genuine customer too.

K-means

Since the proposed method assigns a label of spam, mixed and non-spam to each reviewer, the algorithm is essentially doing clustering. We thus use the most popular clustering algorithm *k*-means as a baseline.

We now present the results of the *k*-means clustering. K-means clustering is a method of cluster analysis which aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean. Since the proposed model produce three clusters, we also let *k*-means to produce three clusters of the 4,465 reviewers. Each reviewer is represented by a vector of four feature values described in the previous section.

By applying *k*-means, we are able to detect 899 spammers and 2,391 non-spammers (the rest are mixed) based on the cluster centroids. We build two classifiers for the reviews written by the reviewers in the spammer and non-spammer clusters. In both classifiers, we treat the reviews written by spammers as positive and reviews written by non-spammers as negative. In the first classifier, we classify all the reviews written by these reviewers including both burst reviews and non-burst reviews. We get 6,493 reviews for spammers and 19,627 reviews for non-spammers and we randomly sample 2,000 reviews from each class for 5-fold cross validation. We use the balanced data, i.e., 50% of the reviews are from spammers and 50% of the reviews are from non-spammers, for classification just to make the results clearer and easier to understand. In the second classifier, we only extract reviews of these reviewers from bursts, and get 1,956 for spammers and 4,728 reviews for non-spammers, and we randomly samples 1,800 reviews and performed 5-fold cross validation. The classification results are shown in Table 2.

Table 2. Classification results from *k*-means clustering

	precision	recall	F-score	accuracy
all reviews	53.2%	66.0%	58.9%	53.9%
burst reviews	55.9%	71.4%	62.7%	57.5%

Since the objective of classification here is to test if reviews of spammers and non-spammers are separable, and the size of positive class and negative class is the same (balanced data), classification accuracy is more important than F-score. From Table 2, we can see that the result of using all reviews is only slightly better than random (which should give us 50% of accuracy). However, using reviews only from bursts can help us achieve slightly better result, which agrees with our hypothesis about the nature of reviewers within bursts. Overall the classification results are quite poor, which indicate that *k*-means clustering is not accurate in identifying spammers and non-spammers.

LBP with State Prior Only

As stated in the previous section, we use a similar approach to that in (Pandit et al. 2007) as a baseline. Although the authors did not use prior, we use state prior as fully unsupervised LBP are known to produce poor models (Yedidia, Freeman, and Weiss 2001). By using only state prior based on ratio of Amazon Verified Purchase and propagation matrix, 278 reviewers are labeled as spammers and 871 reviewers are labeled as non-spammers (the rest are labeled as mixed). Similarly, we build two classifiers for the reviews written by spammers and non-spammers and treat reviews of spammers as positive and reviews of non-spammers as negative. In the first classifier, we extract reviews from both bursts and non-bursts, and we get 2,439 reviews for spammers and 8,270 reviews for non-spammers. Then we randomly sample 2,000 reviews from each class to perform 5-fold cross validation. In the second classifier, we extract each reviewer's reviews only from bursts and we get 742 reviews for spammers and 2,335 reviews for non-spammers, and we randomly sample 700 reviews from each class for 5-fold cross validation. The results of both classifiers are shown in Table 3.

Table 3. Classification results from LBP with prior only

	Precision	recall	F-score	accuracy
all reviews	57.3%	59.3%	58.3%	57.5%
burst reviews	61.2%	55.3%	57.9%	59.6%

From the table, firstly we can see that the classification again shows better result for reviews within bursts than for all reviews of the reviewers. Secondly, comparing with the results from *k*-means, we notice that by modeling the reviewers with MRF and considering the burstiness nature of reviewers in bursts, we do get better accuracy results both for all reviews and for reviews that appear in bursts only, but the improvements are not much. However, for F-score, the results are actually worse for burst reviews. This shows that LBP with state prior is still not effective.

LBP with State Prior and Local Observation

In this setting, we employ the proposed spamming behavior indicator as a local observation for each node and induce the local observation in the message passing of LBP algorithm. By inducing the local observation, the state of a node no longer depends only on the messages sent from its neighboring nodes, but also depends on the observed information in the data.

By involving the local observation, 508 reviewers are labeled as spammers and 794 reviewers are labeled as non-spammers. We again build two classifiers to classify the reviews written by spammers against those written by non-spammers. There are 1,279 reviews of spammers and 1,862 reviews of non-spammers that appear in bursts. We randomly sample 1,000 reviews from each class for 5-fold cross validation. Also, 3,898 reviews of spammers and 6,817 reviews of non-spammers are extracted from both bursts and non-bursts. We randomly sample 2,000 reviews from each class for 5-fold cross validation. Both results are shown in Table 4.

Table 4. Classification results from LBP with prior and local observation

	precision	recall	F-score	accuracy
all reviews	77.8%	61.5%	68.7%	71.2%
burst reviews	83.7%	68.6%	75.4%	77.6%

From the above table, firstly we see that classification result for burst reviews is again better than for all reviews. Secondly, as we incorporate local observation, the classification results for all reviews improve dramatically by 13.7% in accuracy and 10.4% in F-score compared with using only state prior in the model (Table 3). For burst reviews, the improvements are even greater, by around 18% in both accuracy and F-score.

Finally, we note again that there is no overlap between spamming behavior features in detecting spammers and the features used in review classification, which suggests the correlation between spamming behaviors and spam reviews. In the next section, we will use human evaluation to further confirm this correlation and the effectiveness of our model.

Human Evaluation

Our second evaluation is based on human expert judgment, which was commonly used in research on spam, e.g., Web spam (Spirin and Han 2012), email spam (Chirita, Diederich and Nejdl 2005), and even blogs and social spam (Kolari et al. 2006). Human evaluation has also been used for opinion spam in prior works (Lim et al. 2010; Wang et al. 2011; Mukherjee, Liu and Glance 2012; Xie et al. 2012). It is, however, important to note that just by reading a single review without any context, it is very hard to determine whether a review is fake (spam) or not (Jindal and Liu 2008; Ott et al. 2011). However, it has been shown in (Mukherjee, Liu and Glance 2012) that when a context is provided e.g., reviewing patterns, ratings, brand of products reviewed, posting activity trails, etc., human expert evaluation becomes easier.

For this work, we used 3 domain expert judges, employees of an online shopping site, to evaluate our results. The judges had sufficient background knowledge about reviews of products and sellers due to the nature of their work in online shopping. The judges were briefed with many opinion spam signals: i) Having zero caveats, and full of empty adjectives. ii) Purely glowing praises with no downsides. iii) Suspicious brand affinity/aversion, unusual posting activity, etc., from prior findings and consumer sites (Popken 2010; Frietchen 2009). These signals are sensible as they have been compiled by consumer domain experts with extensive know-how on fake reviews. Our judges were also familiar with Amazon reviews and given access to additional metadata, e.g., review profile, demographic information, and helpfulness votes. Although the judges were not provided the proposed features, they were encouraged to use their own signals along with the above existing signals and reviewer metadata. It is important here to note that providing various signals compiled from prior works and domain experts in consumer sites (Popken 2010; Frietchen 2009) to the judges do not introduce a bias but enhances judgment. Without any signals, as mentioned above, it is very difficult to judge by merely reading reviews. It is also hard for anyone to know a large number of signals without extensive experience in opinion spam detection. Given a reviewer and his reviews, the judges were asked to independently examine his entire profile (along with relevant metadata) to provide a label as spammer or non-spammer.

Due to the large number (4,465) of reviewers in our data, it would have taken too much time for human judges to assess all the reviewers in a short time, so we are not able to evaluate the recall of our method. We thus only randomly selected 50 reviewers from spammers and non-spammers detected by each method: LBP without local observation, LBP with local observation and K-means, and gave to our judges for evaluation.

Table 5 reports the result of each judge for spammers and non-spammers (as the count of reviewers judged as spammers out of the 50 reviewers identified as spammers or non-spammers) of each method. Additionally, we report the agreement of judges using Fleiss multi-rater kappa (Fleiss, 1971) for each method in the last row of Table 5.

Table 5: Human judgment results

	k-means		Without local		With local	
	spam	Non-spam	spam	Non-spam	spam	Non-spam
J1	16	14	29	5	41	2
J2	13	9	27	4	36	0
J3	14	12	28	3	37	1
Avg	14.33	11.67	28	4	38	1
Kappa	0.72	0.70	0.69	0.78	0.71	0.84

Since we report our results from human evaluators in terms of the count of reviewers judged as spammers, we expect the count in the non-spam columns to be low. From the table, we can see that *k*-means performs the worst both in terms of detecting spammers and non-spammers. And the results of employing LBP with infused local observation are the best. All the results given by our human judges are consistent with our previous classification results, which also show the effectiveness of using classification as a means of evaluation.

Spammer Detection Example Case

In this sub-section, we take a close look at an example of review spammer detection using the method we proposed in this paper. We investigate the reviewers within the review burst of the product (id: B000TME1HW) in the Amazon data set. Spammers detected by our method are shown to be concentrated in this burst (6 out of 7 reviewers are labeled as spammers[1]). We apply the review burst detection techniques described in the previous section and plot the histogram and kernel density estimation for this product, which are shown in Figure 1.

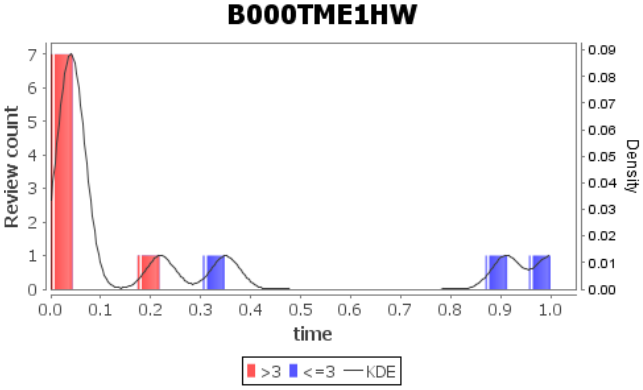

Figure 1. Review histogram and KDE of product ‘B000TME1HW’

In Figure 1, red bars indicate reviews that have ratings greater than 3, blue bars indicate reviews with rating less than or equal to 3, and the curve represents the estimated density of the histogram. Based on the method of burst detection described in the previous section, only the seven reviews in the first bar are considered as burst reviews in this product, which correspond to the reviews from September 28, 2007 to October 10, 2007.

In order to examine the validity of the result produced by our model, we went to the profile page of each reviewer within the burst. By carefully studying their profiles, we have the following observations: (1) 6 out of 7 reviewers within this burst never have reviews marked as “Amazon Verified Purchase”. (2) 6 out of 7 reviewers wrote three or more reviews within a single day; 4 out of 7 reviewers wrote five or more reviews within two days. (3) 6 out of 7 reviewers who appear in this burst also reviewed other products of the same brand (may not be the same products) around the same time, some reviewers posted exactly the same reviews, and others posted different reviews to these products.

Furthermore, by looking at the reviewers’ arrival pattern, we feel that the reviewers within the burst are suspicious. Almost all the good reviews fall into this burst and they all arrived together. However, all bad reviews (all with 1 star) arrived in a random pattern afterwards. Based on our intuition, good reviews and bad reviews should be mixed together, and the arrival pattern of good reviews should not be so concentrated. All these observations make us feel confident that these reviewers are spammers.

Although we test our method using Amazon reviews, the idea and the method can also be applied to other review hosting sites to detect review spammers with only minor changes. As we mentioned before, a generic framework that can deal with both reviewers with multiple reviews and a single review is considered as our future work.

Conclusion

In this paper, we proposed to exploit bursts in detecting opinion spammers due to the similar nature of reviewers in a burst. A graph propagation method for identifying spammers was presented. A novel evaluation method based on supervised learning was also described to deal with the difficult problem of evaluation without ground truth data, which classifies reviews based on a different set of features from the features used in identifying spammers. Our experimental results using Amazon.com reviews from the software domain showed that the proposed method is effective, which not only demonstrated its effectiveness objectively based on supervised learning (or classification), but also subjectively based on human expert evaluation. The fact that the supervised learning/classification results are consistent with human judgment also indicates that the proposed supervised learning based evaluation technique is justified.

Acknowledgements

This project is supported in part by a grant from HP labs Innovation Research Program and by a grant from National Science Foundation (NSF) under grant no. IIS-1111092.

[1] http://www.amazon.com/gp/pdp/profile/A11LLS9F0SYXJW/ref=cm_cr_pr_pdp
http://www.amazon.com/gp/pdp/profile/A3A7H7WW2BKTMV/ref=cm_cr_pr_pdp
http://www.amazon.com/gp/pdp/profile/A2NE89LGFA3EP/ref=cm_cr_pr_pdp
http://www.amazon.com/gp/pdp/profile/A2XCWGUK30L1C/ref=cm_cr_pr_pdp
http://www.amazon.com/gp/pdp/profile/A1B7KMWDJ1886U/ref=cm_cr_pr_pdp
http://www.amazon.com/gp/pdp/profile/A3NEEVREFZSUER/ref=cm_cr_pr_pdp

References

Chirita, P.-A., Diederich, J., and Nejdl, W. 2005. MailRank: using ranking for spam detection. In proceedings of the 14th ACM international conference on Information and knowledge management, CIKM ’05, 373-380. New York, NY, USA: ACM.

Duda, R. O., Hart, P. E., and Stork, D. G. Pattern Classification. New York: Wiley, 2000.

Feng, S., Xing, L., Gogar, A., and Choi, Y. 2012. Distributional Footprints of Deceptive Product Reviews. In proceedings of the Sixth International AAAI Conference on Weblogs and Social Media, 98-105, Dublin, Ireland: AAAI Press.

Frietchen, C. 2009. How to spot fake user reviews. consumersearch.com.

Fleiss, J. L. 1971. Measuring nominal scale agreement among any raters. Psychological Bulletin: 378-382.

Jindal, N., Liu, B., and Lim, E.-P. 2010. Finding unusual review patterns using unexpected rules. In Proceedings of the 19th ACM Conference on Information and Knowledge Management, 1549–1552.

Jindal, N., and Liu, B. 2008. Opinion spam and analysis. In Proceedings of the international conference on Web search and web data mining, WSDM ’08, 219–230. New York, NY, USA: ACM.

Kolari, P., Java, A., Finin, T., Oates, T., and Joshi, A. 2006. Detecting spam blogs: a machine learning approach. In proceedings of the 21st national conference on Artificial intelligence - Volume 2, 1351-1356, AAAI Press.

Li, F., Huang, M., Yang, Y., and Zhu, X. 2011. Learning to identify review spam. In proceedings of the 22nd international joint conference on Artificial Intelligence - Volume Three, 2488-2493, AAAI Press.

Lim, E.-P., Nguyen, V.-A., Jindal, N., Liu, B., and Lauw, H. W. 2010. Detecting product review spammers using rating behaviors. In Proceedings of the 19th ACM international conference on Information and knowledge management, CIKM ’10, 939–948. New York, NY, USA: ACM.

Mukherjee, A., Venkataraman, V., Liu, B., and Glance, N. 2013. What Yelp Fake Review Filter might be Doing? A Case Study on Commercial vs. Crowdsourced Fake Reviews. To appear in Proceedings of the 7th International AAAI Conference on Weblogs and Social Media (ICWSM).

Mukherjee, A., Liu, B., Wang, J., Glance, N. S. 2012. Spotting fake reviewer groups in consumer reviews. In Proceedings of the 21st international conference on World Wide Web, WWW ’12, 191-200, New York, NY, USA: ACM.

Ott, M., Choi, Y., Cardie, C., and Hancock, J. T. 2011. Finding deceptive opinion spam by any stretch of the imagination. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 309–319. Portland, Oregon, USA: Association for Computational Linguistics.

Pandit, S., Chau, D. H., Wang, S., and Faloutsos, C. 2007. NetProbe: a fast and scalable system for fraud detection in online auction networks. In proceedings of the 16th international conference on World Wide Web, WWW ’07, 201-210, New York, NY, USA: ACM.

Popken, B. 2010. 30 Ways You Can Spot Fake Online Reviews. The Consumerist.

Scott, D. W. Multivariate Density Estimation. New York: Wiley-InterScience, 1992.

Spirin, N., and Han, J. 2012. Survey on web spam detection: principles and algorithms. ACM SIGKDD Explorations. Volume 13 Issue 2, 50-64.

Wang, G., Xie, S., Liu, B., and Yu, P. S. 2011. Review Graph Based Online Store Review Spammer Detection. In proceeding of the 11th IEEE International Conference on Data Mining, ICDM ’11, 1242-1247, Vancouver, BC, Canada: IEEE.

Xie, S., Wang, G., Lin, S., Yu, P. S. 2012. Review spam detection via temporal pattern discovery. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD ’12, 823-831, New York, NY, USA: ACM.

Yedidia, J. S., Freeman, W. T., and Weiss, Y. Understanding belief propagation and its generalizations. Exploring artificial intelligence in the new millennium, 239 – 269.