Opinion Spam Detection: An Unsupervised Approach using Generative Models

Arjun Mukherjee
Department of Computer Science
University of Houston
501 Philip G. Hoffman Hall (PGH), 4800
Calhoun Rd. Houston, TX 77204-3010
arjun@cs.uh.edu

Vivek Venkataraman
Department of Computer Science
University of Illinois at Chicago
851 S Morgan St. Chicago, IL 60607
vivek1186@gmail.com

Abstract

Opinionated social media such as consumer reviews are widely used for decision making. However, due to the reason of profit or fame, imposters have tried to game the system by opinion spamming (e.g., writing deceptive fake reviews) to promote or to demote some target entities. In recent years, opinion spam detection has attracted significant attention from both industry and academic research. Most existing works on opinion spam detection are supervised and/or rely on heuristics. However, prior works have shown that obtaining large scale and reliable labels to serve as training data is nontrivial, costly, time consuming, and usually requires domain expertise. Thus, the problem remains to be highly challenging. This paper proposes an unsupervised approach for opinion spam detection. A novel generative model for deception is proposed which can exploit both linguistic and behavioral footprints left behind by spammers. Experiments using three real-world opinion spam datasets demonstrate the effectiveness of the proposed approach which significantly outperforms strong baselines. The estimated language models also render insights into the language aspects of deceptive opinions on the Web.

Public opinion in this country is everything.

—Abraham Lincoln

1 Introduction

Opinions have come a long way. Nowadays, almost everyone views online reviews before deciding on a restaurant, hotel, buying a product, or even choosing a travel destination. Consumer opinions have escalated to stature of a valuable resource for decision making. However, with its usefulness, it brings forth a curse — *deceptive opinion spam*. As positive/negative opinions directly translate to significant financial gains/losses for businesses, imposters try to game the system by posting deceptive fake reviews to promote or to discredit target entities (e.g., products, businesses, services, etc.). Such activities are called opinion spamming. The imposters are called *opinion spammers* or *fake reviewers*. As more and more individuals and organizations are using reviews for their decision making, detecting opinion spam has become a pressing issue. The problem has been widely reported in the news (Streitfeld, 2012).

First studied in (Jindal and Liu, 2008), it has attracted significant interest in recent years. Several dimensions of the problem have been explored ranging from detecting individual (Lim et al., 2010) and group (Mukherjee et al., 2012) opinion spammers, to detecting deceptive opinions in reviews (Li et al., 2011; Ott et al., 2011) to time-series (Xie et al., 2012), deception prevalence (Ott et al., 2012), stylometric (Feng et al., 2012a), and distributional (Feng et al., 2012b) analyses. These approaches have primarily focused on supervised learning. However, obtaining reliable labeled data for training is nontrivial. The two main successful approaches are: (1) Ott et al., (2011) who gathered fake reviews using Amazon Mechanical Turk (AMT) crowdsourcing tool, and (2) Mukherjee et al., (2012) who employed domain experts to produce a labeled dataset of fake reviewers. However, both these approaches are expensive and painstaking posing a problem for large scale machine learning and analysis.

In this paper, we propose a novel and principled unsupervised modeling technique to detect opinion spam in the Bayesian setting. We formulate opinion spam detection as a Bayesian clustering problem. The Bayesian setting allows us to elegantly model "spamicity" (degree of spamming) of authors and reviews as latent variables with other observed behavioral and linguistic features in our Latent Spam

Model (LSM). Although LSM estimates both author (reviewer) spamicity and whether a review is spam (fake) or non-spam (non-fake), in this work, we focus on fake review detection. The intuition behind LSM hinges on the hypothesis that opinion spammers differ from others (non-spammers) on linguistic and behavioral dimensions (Ott et al., 2011; Lim et al., 2010). This creates a separation margin between population distributions of two naturally occurring clusters: spam vs. non-spam. LSM aims to learn the population distributions of two classes. This paper makes the following main contributions:

1. A novel unsupervised generative model is proposed for detecting opinion spam exploiting linguistic and behavioral features of authors and reviews. The model is very general and can be applied to almost any review hosting site having sufficient metadata.
2. Two variations of the model is proposed leveraging different kinds of priors.
3. The proposed model is evaluated on three labeled real-world opinion spam datasets. Experimental results show that the proposed method outperforms state-of-the-art baselines significantly across all datasets.
4. The posterior estimates of the latent variables of the model also render insights into some language aspects of deceptive opinions on the Web. To our knowledge such an investigation has not been done before.

2 Related Work

Beyond the previous works mentioned in §1, several other dimensions have also been explored in opinion spam. In (Jindal et al., 2010), different reviewing patterns were discovered by mining unexpected class association rules. In (Lim et al., 2010), some behavioral patterns were designed to rank reviewers. In (Wang et al., 2011), a graph-based method for ranking store spam reviewers was proposed. Fei et al., (2013) explored burstiness patterns in reviews and in (Mukherjee et al., 2013) distributional divergence of abnormal behaviors were investigated. There have also been dedicated studies on negative opinion spam (Ott et al., 2013) and exploiting product profiles (Feng and Hirst, 2013). Although all these approaches have made important progresses, they are, however, mostly supervised and/or are based on heuristics or human observations. To our knowledge, no principled models combining both behavioral and linguistic characteristics in the unsupervised setting have been proposed so far which is the main focus of this work.

In a wide field, a study of bias, controversy and summarization of research paper reviews was also reported in (Lauw et al., 2006; 2007). However, this is a different problem as research paper reviews do not (at least not obviously) involve faking. Studies on review quality (Liu et al., 2007), distortion (Wu et al., 2010), and helpfulness (Danescu-Niculescu-Mizil et al., 2009; Kim et al., 2006) were also conducted. These works do not detect fake reviews.

Spam has been widely investigated on the Web (Spirin and Han, 2012; Lee and Ng, 2005; and references therein) and email networks (Sahami et al., 1998). Recent studies on spam also extended to blogs (Kolari et al., 2006), online tagging (Koutrika et al., 2007), clickbots and bot generated search traffic (Yu et al., 2010), and social networks (Jin et al., 2011). However, the dynamics of all these forms of spamming are quite different from those of deceptive opinion spam in reviews. Unlike opinion spam, most other spam activities usually involve commercial advertising which makes them slightly easier to detect. Online reviews, on the other hand, seldom contain commercial advertising.

Also related is the task of psycholinguistic deception detection which investigates lying words (Hancock et al., 2008; Newman et al. 2003), untrue views (Mihalcea and Strapparava (2009), computer-mediated deception in role-playing games (Zhou et al., 2008), etc. These works mostly study deception from a qualitative and psycholinguistic perspective and/or use supervised learning. Our focus is unsupervised detection of deceptive fake reviews in online reviews sites.

3 Model

We now detail our proposed model. We first discuss the basic intuition (§3.1) and the observed features (§3.2), and then propose the generative process of our model (§3.3). Finally, we detail inference methods in §3.4 and §3.5.

3.1 Intuition and Overview

We model fake review detection as an instance of unsupervised Bayesian clustering with two clusters, spam and non-spam. The Bayesian setting conveniently allows us to treat spamicity of authors/reviews as latent variables in our model. Specifically, we model the spam/non-spam category of a review as a latent variable π (See Table 1). This can be seen as the category/class variable reflecting the cluster memberships of every review.

The proposed Latent Spam Model (LSM) belongs to the class of generative models for clustering (Duda et al., 2001). Each review of an author is represented with a set of observed linguistic and behavioral features which are emitted conditioned on the latent spam/non-spam category variable and associated distributions. The goal is to learn the latent category assignments for each review and the per-category distributions. This is achieved using posterior inference techniques (e.g., Markov Chain Monte Carlo) for probabilistic model-based clustering (Smyth, 1999). The stationary distributions of class/category assignments is used for generating clusters of spam (fake) and non-spam (non-fake) reviews.

3.2 Observed Features

Linguistic n-grams have been showed to be useful for deception detection (Ott et al., 2011). Thus, we use words (unigrams)[1] as our linguistic features. Our behavioral features are constructed from various abnormal behavioral patterns of reviewers and reviews. We first list the author (reviewer) features and then the review features. The notations are listed in Table 1.

Author Features: The proposed continuous author features in [0, 1] are listed below. Values close to 0/1 indicate non-spamming/spamming respectively.

1. Content Similarity (CS): Spammers typically post fake experiences. However, as crafting a new fake review every time is time consuming, they often post reviews which are duplicate/near-duplicate versions of their previous reviews (Jindal and Liu, 2008). It is naturally useful to capture the maximum content similarity (using cosine similarity) across any pair of reviews by an author/reviewer, a. We use the maximum similarity to capture the worst spamming behavior.

$$f_{CS}(a) = \max_{r_i, r_j \in R_a, i<j} cosine(r_i, r_j) \qquad (1)$$

2. Maximum Number of Reviews (MNR): Posting many reviews in a single day reflects abnormal reviewing pattern and can be used as a behavioral feature. This feature simply computes the maximum number of reviews posted in a day for an author. It is normalized by the maximum value in the dataset.

Variable/Function	Description
$a;\ A;\ r$	An author a; set of all authors; a review
$r_a = (a, r)$; R_a	Review/All reviews by a, $R_a = \{r_a\}$
$b(r_a)$; $\star\,(r_a, b(r_a))$	Entity b of r_a; $\star$ rating of r_a on $b(r_a)$.
$MaxRev(a)$	Max # of reviews in a day by author, a
$F(a)$; $L(a)$	First/Last posting date of author, a
$dt(a, b)$	Review date of author a on entity b
$A(b)$	Launch date of entity b
$s_a \sim Beta(\alpha_{\hat{s}}, \alpha_{\hat{n}})$	Spamicity of an author, a, $s_a \in\ [0, 1]$
$\alpha^a_{k \in \{\hat{s}, \hat{n}\}}$	Beta priors for s_a for author a
$k \in \{\hat{s}, \hat{n}\}$	Class variable $k \in$ {Spam, Non-spam}
$\pi_{r_a} \in \{\hat{s}, \hat{n}\}$	Class variable for review r_a
$\theta^f_{k \in \{\hat{s}, \hat{n}\}} \sim Beta(\gamma^f_{\hat{s}}, \gamma^f_{\hat{n}})$	Class prob. of $f \in \{EXT, DEV, ETF\}$
$\gamma^f_{k \in \{\hat{s}, \hat{n}\}}$	Beta priors of θ^f for review behavior, f
$\psi^f_{k \in \{\hat{s}, \hat{n}\}} \sim Beta$	Per class prob. of $f \in \{CS, MNR, ACT\}$
$\varphi_{k \in \{\hat{s}, \hat{n}\}} \sim\ Dir(\beta)$	Multinomial of words for class k
$w_{a,r,j}$; $N_{a,r}$	j^{th} word; total words in r by a
$x^f_{r_a} \sim\ Bern(\theta^f_k)$	Observed $f \in \{EXT, DEV, ETF\}$of r_a
$y^f_{r_a} \sim \psi^f_k$	Observed $f \in \{CS, MNR, ACT\}$ of r_a
V	Vocabulary (set of all words).
$W_{i,v}$	# word v appears in review i
$n^W_{k,v}$	# word v appears in reviews of class k
$n_{a,\hat{s}}$; $n_{a,\hat{n}}$	# reviews of a assigned spam; non-spam
$n^f_{k,P}$; $n^f_{k,A}$	# reviews in class k with $f = 0/1$ present; absent
n_a ; $n_{k \in \{\hat{s}, \hat{n}\}}$	# reviews by a; # reviews in class k
K	Total categories/classes in the model

Table 1: List of notations

$$f_{MNR}(a) = \frac{MaxRev(a)}{\max\limits_{a \in A}(MaxRev(a))} \qquad (2)$$

3. Reviewing Activity (ACT): The study in Lim et al., (2010) reports that opinion spammers are usually not longtime members of a site. Genuine reviewers, however, use their accounts from time to time to post reviews over a considerably long period of time. It is thus useful to exploit the activity freshness of an account to detect spamming. We compute the activity of an author by measuring the difference of his first and last review posting dates. We normalize this feature by the maximum value in our dataset. This activity feature indicates that authors posting reviews over a reasonably long time frame are less likely to be spamming than those who just created their accounts to some post specific (probably deceptive/spam) reviews and do not ever use that account afterwards.

$$f_{ACT}(a) = 1 - \frac{L(a) - F(a)}{\max\limits_{a \in A}(L(a) - F(a))} \qquad (3)$$

[1] Higher order n-grams did not improve model performance.

Review Features: We now propose three boolean[2] review features which can be used as indicators. Feature values close to 0/1 indicate non-spamming/spamming respectively.

4. Extreme Rating ($\boldsymbol{EXT}$): Opinion spamming typically projects entities incorrectly either in a very positive or a very negative light (Jindal and Liu, 2008). Thus, on a 5-star rating scale, spammers are likely to give extreme ratings (1 or 5 stars) in order to promote or to demote entities. The following review feature accounts whether the associated star rating of the review was extreme or not.

$$f_{EXT}(r_a) = \begin{cases} 1,\star\left(r_a, b(r_a)\right) \in \{1,5\} \\ 0,\star\left(r_a, b(r_a)\right) \in \{2,3,4\} \end{cases} \tag{4}$$

5. Rating Deviation $(\boldsymbol{DEV})$: As noted above, opinion spamming usually involves incorrect projection either in the positive or negative light so as to alter the true sentiment on the entity. Naturally, this reflects the intuition that the ratings of spammers should be deviating from the average ratings given by other genuine reviewers. On a 5-star scale, the absolute rating deviation of a review from the general rating consensus (average rating of the entity) can be between 0 and 4. As different reviewers have different rating levels, we set a reasonable threshold of $\delta = 2$ points to indicate pronounced deviation from the rating consensus on that entity.

$$f_{DEV}(r_a) = \begin{cases} 1, \left|\star\left(r_a, b(r_a)\right) - E\left[\star\left(r_{a' \neq a}, b(r_a)\right)\right]\right| \geq \delta \\ \quad 0, otherwise \end{cases} \tag{5}$$

The second expectation term is taken over all reviews on an entity $b = b(r_a)$ by other authors, $a' \neq a$ to obtain the average rating consensus on b.

6. Early Time Frame ($\boldsymbol{ETF}$): Lim et al. (2010) noted that spammers often review early to inflict spam as the early reviews can greatly impact the people's sentiment on the entity. To capture this spamming characteristic, we measure whether a review is posted within some early time frame.

$$f_{ETF}(r_a) = ETF(r_a, b = b(r_a)) = \begin{cases} 1, dt(a,b) - A(b) < \tau \\ \quad 0, otherwise \end{cases} \tag{6}$$

$ETF(r_a, b)$ captures whether an author, a reviewed the entity b early enough with respect to its launch date, $A(b)$. For websites where the launch date of a product/business is not publicly available, it can be approximated by the first date of review on that entity. Following prior work in (Mukherjee et al., 2012), we set τ = 9 months as a threshold for denoting earliness.

It is important here to note that the thresholds values (δ = 2, τ = 9) were set following the heuristics suggested in prior works (Lim et al., 2010; Mukherjee et al., 2012). Ideally, they should be learned from the data. However, this work focuses on detecting opinion spam in the fully unsupervised setting. Hence, we cannot use any supervised information to learn the threshold values using cross validation. Our key contribution in this work is to propose a principled model for fake review detection.

Lastly, we note that there may be various other pieces of metadata about authors/reviewers that are specific to the domain or website and may be useful in modeling opinion spam. For example, in Amazon, reviewers obtain badges, ranks, verified purchase tags, etc., while in Yelp, there are several social networking metadata of reviewers (e.g., friendship and fan relations, compliments, tip counts, etc.). However, using these features for model building can affect the generalization capabilities of our model as they are website specific. Hence, we do not use them in our present framework. However, as we will see in the subsequent sections, our model is flexible and any number of additional features can be easily added for appropriate augmentations.

3.3 Process

This section explain the generative process of LSM. We refer to notations in Table 1 for the following description. In LSM, the spam/non-spam (fake/non-fake) cluster discovery is influenced by linguistic features (words/unigrams) in reviews and observed behavioral features mentioned in §3.2. We model (normalized) continuous values of author features as Beta distributions which allow LSM to see more fine grained dependencies of authors' behaviors with spamming $(y^f_{r_a} \sim \psi^f_k)$. However, review features being more "objective", we found that review spamming behaviors are better captured when they are modeled as binary variables emitted from a Bernoulli distribution, $x^f_{r_a} \sim Bern(\theta^f_k)$. In LSM (Figure 1), we have K= 2 categories/classes, $k \in \{\hat{s}, \hat{n}\}$ (spam/non-spam). $\theta^f_{k \in \{\hat{s}, \hat{n}\}}$ denotes the latent behavioral distribution corresponding to each observed review feature $f \in \{EXT, DEV, ETF\}$ and $\psi^f_{k \in \{\hat{s}, \hat{n}\}}$

[2] Pilot experiments showed that abnormal behavioral patterns in reviews being more objective are better captured using boolean features. See more modeling details in §3.3.

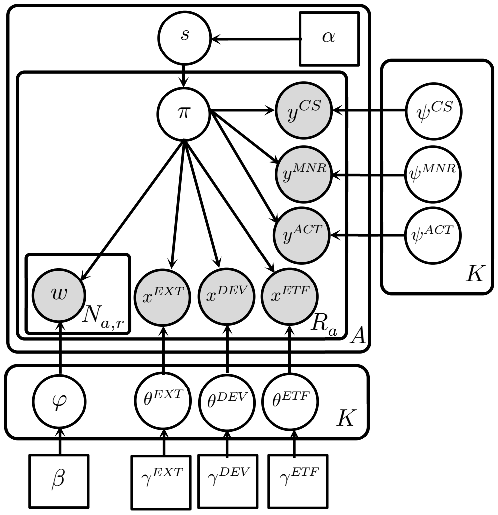


Figure 1: Plate notation of LSM

Algorithm 1 Inference using MCMC Gibbs Sampling

```
1. Initialization:
    Sample $\varphi_k \sim Dir(\beta)$
    Randomly assign review categories, $\pi_{r_a} = \begin{cases} \hat{s}, z < 0.5 \\ \hat{n}, z \geq 0.5 \end{cases}; z \sim U(0,1)$
2. Iterate $n = 1$ to $N_{max}$:
    For author, $a = 1$ to $A$:
        For review $r_a = 1$ to $R_a$:
            i. Flush $\pi_{r_a}$ class assignment
            ii. Sample $\pi_{r_a} \sim p(\pi = k| \dots)$ using (7)
            iii. Update $n_{k,[\,]}^{W}, n_{k,[\,]}^{f=EXT}, n_{k,[\,]}^{f=DEV}, n_{k,[\,]}^{f=ETF}$ for $k \in \{\hat{s}, \hat{n}\}$
        End for
    End for
    Sample $\varphi_k \sim Dir(n_{k,[\,]}^{W} + \beta)$
    If $n > N_{BurnIn}$:
        For author, $a = 1$ to $A$:
            For review $r_a = 1$ to $R_a$:
                i. Update $\psi_k^{f=CS}, \psi_k^{f=MNR}, \psi_k^{f=ACT}$; $k \in \{\hat{s}, \hat{n}\}$ using (10)
            End for
        End for
    End if
```

denotes the latent behavioral distribution corresponding to each observed author feature $f \in \{CS, MNR, ACT\}$. Additionally, s_a denotes the spamicity of an author, a on the scale [0, 1] (where values close to 0/1 indicate non-spamming/spamming). π_{r_a} denotes the class, $k \in \{\hat{s}, \hat{n}\}$ for review, r_a authored by author a. $w_{a,r,j}$, $N_{a,r}$ denote the j^{th} word and the total number of words in review r_a respectively. $\varphi_{k \in \{\hat{s}, \hat{n}\}}$ denotes the word distribution for spam and non-spam review category. The generative process of LSM is given below.

1. For each category, $k \in \{\hat{s}, \hat{n}\}$:
    Draw $\varphi_k \sim Dir(\beta)$; Draw $\theta_k^{f \in \{EXT,DEV,ETF\}} \sim Beta(\gamma_k^f)$
2. For each reviewer (author) $a \in \{1 \dots A\}$:
    i. Draw author spamicity, $s_a \sim Beta(\alpha^a)$
    ii. For each review, $r_a \in \{1 \dots R_a\}$:
        a. Draw its category, $\pi_{r_a} \sim Bern(s_a)$
        b. Emit review features $f \in \{EXT, DEV, ETF\}$:
            $x_{r_a}^f \sim Bern\left(\theta_{\pi_{r_a}}^f\right)$
        c. Emit author features $f \in \{CS, MNR, ACT\}$:
            $y_{r_a}^f \sim \psi_{\pi_{r_a}}^f$
        d. For each word j in review r_a, $j \in \{1 \dots N_{a,r}\}$:
            Emit $w_{a,r,j} \sim Mult\left(\varphi_{\pi_{r_a}}\right)$

Lastly, we note that observed author features are placed in the review plate (Figure 1). This is reasonable because each author behavior can be thought of as percolating through various reviews of that author and emitted across each review to some extent. Doing this renders two key advantages: i) It permits us to exploit a larger co-occurrence domain, which often results in more robust models. ii) It yields a simplified sampling distribution providing for faster inference.

3.4 Inference

To learn the model, we resort to approximate posterior inference using MCMC Gibbs sampling. We employ Rao-Blackwellization (Bishop, 2006) to reduce sampling variance by collapsing latent variables s and θ^f. For observed author features, since we use continuous Beta distributions, sparsity is considerably less and not a big concern here as far as parameter estimation of ψ^f is concerned. To ensure efficient inference, we estimate ψ_k^f using the method of moments, once per sweep of Gibbs sampling. The Gibbs sampler is given by:

$$p(\pi_i = k|\pi_{\neg i} \dots) \propto \prod_{v=1}^{V} (\varphi_{k,v})^{W_{i,v}} \frac{n_{a,k_{\neg i}} + \alpha_k^a}{(n_a + \alpha_{\hat{s}}^a + \alpha_{\hat{n}}^a)_{\neg i}} \times \prod_{f \in \{EXT,DEV,ETF\}} \left(g(f, k, x_{a,r}^f)\right) \times \prod_{f \in \{CS,MNR,ACT\}} \left(p\left(y_{a,r}^f | \psi_{\pi_i}^f\right)\right) \tag{7}$$

where the function g and $p\left(y_{a,r}^f | \psi_{\pi_i}\right)$ are given by:

$$g\left(f, k, x_{a,r}^f\right) = \begin{cases} \frac{\left(n_{k,P}^f + \gamma_k^f\right)_{\neg i}}{\left(n_k + \gamma_{\hat{s}}^f + \gamma_{\hat{n}}^f\right)_{\neg i}}, & if \quad x_{a,r}^f = 1 \\ \frac{\left(n_{k,A}^f + \gamma_{\neg k}^f\right)_{\neg i}}{\left(n_k + \gamma_{\hat{s}}^f + \gamma_{\hat{n}}^f\right)_{\neg i}}, & if \quad x_{a,r}^f = 0 \end{cases} \tag{8}$$

$$p\left(y_{a,r}^f | \psi_{\pi_i}\right) \propto \left(y_{a,r}^f\right)^{\psi_{\pi_{i\hat{s}}} - 1} \left(1 - y_{a,r}^f\right)^{\psi_{\pi_{i\hat{n}}} - 1} \tag{9}$$

The subscript $\neg i$ denotes counts excluding review $i = (a, r) = r_a$. Parameter updates for ψ_k^f are given as follows:

$$\psi_k^f = (\psi_{k,\hat{s}}^f, \psi_{k,\hat{n}}^f)$$

$$= \left(\mu_k^f \left(\frac{\mu_k^f(1-\mu_k^f)}{\sigma_k^f} - 1\right), \left(1-\mu_k^f\right)\left(\frac{\mu_k^f(1-\mu_k^f)}{\sigma_k^f} - 1\right)\right) \quad (10)$$

where μ_k^f and σ_k^f denote the mean and biased sample variance for feature f corresponding to class k. Algorithm 1 details the full inference procedure. Omission of a latter index denoted by [ ] (Algorithm 1) corresponds to the row vector of the counts spanning over the latter index.

3.5 Hyperparameter Estimation using MCEM

In our preliminary experiments, we found that LSM is not very sensitive to β but sensitive to the hyperparameters α and γ. This is because the hyperparameter β is associated with the language models of fake/non-fake reviews, φ which acts more like a smoothing parameter. Hence it is not very sensitive and values of $\beta < 1$ worked well. However, the hyperparameters α and γ being priors for author spamicity and latent review behaviors, they directly affect spam/non-spam category assignment to reviews. This section details the estimation of hyperparameters α and γ using Monte Carlo EM. We use single sample Monte Carlo EM to learn α and γ (Algorithm 2). The single-sample method is recommended by Celeux et al. (1996) as it is both computationally efficient and often outperforms multiple-sample Monte Carlo EM.

Algorithm 2 learns hyperparameters α and γ which maximize the model's complete log-likelihood, $\mathcal{L}$. We employ an L-BFGS optimizer (Zhu et al., 1997) for maximization. L-BFGS is a quasi-Newton method which does not require the Hessian matrix of second order derivatives. It approximates the Hessian using rank-one updates of first order gradient. A careful observation of the model's complete log-likelihood shows that it is a separable function in α and γ allowing the hyperparameters to be maximized independently. Owing to space constraints, we only provide the final update equations:

$$\alpha_k^a = \underset{\alpha_k^a}{\operatorname{argmax}} \begin{pmatrix} \log\Gamma(\alpha_{\hat{s}}^a + \alpha_{\hat{n}}^a) + \log\Gamma(\alpha_{\hat{s}}^a + n_{a,\hat{s}}) + \log\Gamma(\alpha_{\hat{n}}^a + n_{a,\hat{n}}) \\ -\log\Gamma(\alpha_{\hat{s}}^a) - \log\Gamma(\alpha_{\hat{n}}^a) - \log\Gamma(n_a + \alpha_{\hat{s}}^a + \alpha_{\hat{n}}^a) \end{pmatrix}$$

$$\frac{\partial\mathcal{L}}{\partial\alpha_k^a} = \Psi(\alpha_{\hat{s}}^a + \alpha_{\hat{n}}^a) + \Psi(\alpha_k^a + n_{a,k}) - \Psi(\alpha_k^a) - \Psi(n_a + \alpha_{\hat{s}}^a + \alpha_{\hat{n}}^a) \quad (11)$$

$$\gamma_k^f = \underset{\gamma_k^f}{\operatorname{argmax}} \begin{pmatrix} \log\Gamma(\gamma_{\hat{s}}^f + \gamma_{\hat{n}}^f) + \log\Gamma(\gamma_{\hat{s}}^f + n_{k,P}^f) + \log\Gamma(\gamma_{\hat{n}}^f + n_{k,A}^f) \\ -\log\Gamma(\gamma_{\hat{s}}^f) - \log\Gamma(\gamma_{\hat{n}}^f) - \log\Gamma(n_k + \gamma_{\hat{s}}^f + \gamma_{\hat{n}}^f) \end{pmatrix}$$

$$\frac{\partial\mathcal{L}}{\partial\gamma_{\hat{s}}^f} = \Psi(\gamma_{\hat{s}}^f + \gamma_{\hat{n}}^f) + \Psi(\gamma_{\hat{s}}^f + n_{\hat{s},P}^f) - \Psi(\gamma_{\hat{s}}^f) - \Psi(n_k + \gamma_{\hat{s}}^f + \gamma_{\hat{n}}^f)$$

$$\frac{\partial\mathcal{L}}{\partial\gamma_{\hat{n}}^f} = \Psi(\gamma_{\hat{s}}^f + \gamma_{\hat{n}}^f) + \Psi(\gamma_{\hat{n}}^f + n_{\hat{n},A}^f) - \Psi(\gamma_{\hat{n}}^f) - \Psi(n_k + \gamma_{\hat{s}}^f + \gamma_{\hat{n}}^f) \quad (12)$$

where, $\Psi(\cdot)$ denotes the digamma function.

Algorithm 2 Single-sample Monte Carlo EM

```
1. Initialization:
    Start with uninformed priors: $\alpha^a \leftarrow (1,1)$; $\gamma^f \leftarrow (1,1)$
2. Repeat:
    i.  Run Gibbs sampling to steady state (Algorithm 1) using
        current values of $\alpha^a$, $\gamma^f$.
    ii. Optimize $\alpha^a$ using (11) and $\gamma^f$ using (12)
  Until convergence of $\alpha^a$, $\gamma^f$
```

4 Experiments

We now evaluate our proposed model. Below we first describe our datasets followed by baselines, evaluations metrics, and experimental results.

4.1 Datasets

To evaluate our proposed model, we consider the following labeled datasets for fake review detection.

AMT Dataset (Ott et al., 2011): This dataset contains 400 truthful (non-fake) reviews obtained from Tripadvisor.com across 20 most popular Chicago hotels. 400 deceptive fake reviews were manufactured using Amazon Mechanical Turk (AMT). Turkers (online workers) were asked to write fake reviews assuming they work for the marketing department by portraying the hotel in the positive light. Each Turker wrote one such fake review. The 400 fake reviews were evenly distributed across the same 20 Chicago hotels. Although this dataset has been regarded as a *gold-standard* in (Ott et al., 2011), it lacks behavior information for Turkers. Although the non-fake reviews from Tripadvisor have some behavior information, using behaviors for only non-fake class makes the data asymmetric for clustering. Hence, we only use linguistic features for this data.

Amazon Dataset (Mukherjee et al., 2012): Mukherjee et al., (2012) generated a domain expert labeled dataset of fake reviewer groups for Amazon.com products. The data contains labeled spamicity scores (in the range [0, 1] with 0 indicating non-spam and 1 indicates spam) for 2431 reviewer groups containing 826 distinct reviewers. For each reviewer, we first computed its spamicity score by taking the expectation over all groups to which it belonged. This rendered a spamicity score for each reviewer in the range [0, 1]. The experiments in (Mukherjee et al., 2012) report thresholds values greater than 0.7 indicate marked spam activities. Hence, we use a threshold of $\boldsymbol{\xi}$ = 0.75 in the scale of [0, 1] to obtain spam (respectively non-spam) reviews posted by reviewers having spamicity $> \boldsymbol{\xi}$ ($<$

Algorithm	Feat.	E	P	Prec.	Rec.	F1
K-means (KM)	L	0.99	0.54	52.6	81.2	63.8
	B	–	–	–	–	–
	L+B	–	–	–	–	–
Single-Link HC	L	0.99	0.53	48.3	79.3	60.0
	B	–	–	–	–	–
	L+B	–	–	–	–	–
Complete-Link HC	L	0.99	0.51	49.6	83.1	62.1
	B	–	–	–	–	–
	L+B	–	–	–	–	–
LSM-UP	L+B	0.85	**0.70**	66.0	86.1	**74.6**
LSM-HE	L+B	0.83	**0.72**	66.1	89.0	**75.9**

(a) AMT Dataset (Ott et al., 2011)

E	P	Prec.	Rec.	F1
0.99	0.54	46.3	83.1	59.5
0.99	0.52	47.6	85.0	61.0
0.99	0.52	48.1	85.4	61.5
0.99	0.54	46.1	87.2	60.3
0.99	0.54	46.3	88.0	60.6
0.99	0.54	46.5	88.4	60.9
0.99	0.52	48.1	85.4	61.5
0.99	0.52	48.4	85.6	61.8
0.99	0.52	49.1	85.9	62.5
0.91	**0.63**	57.2	87.7	**69.2**
0.83	**0.70**	63.7	89.2	**74.3**

(b) Amazon (Mukherjee et al., 2012)

E	P	Prec.	Rec.	F1
0.99	0.52	48.0	54.1	50.8
0.99	0.52	48.1	54.2	50.9
0.99	0.51	48.9	55.5	52.0
0.99	0.54	45.5	53.5	49.2
0.99	0.55	45.9	54.0	49.6
0.99	0.55	46.0	55.3	50.2
0.99	0.52	47.5	54.6	50.8
0.99	0.52	48.2	54.9	51.3
0.99	0.52	48.6	55.2	51.7
0.98	**0.56**	55.0	62.6	**58.4**
0.97	**0.60**	59.2	64.1	**61.6**

(c) Yelp Restaurant Dataset

Table 2: Clustering performance comparison on various metrics: entropy (E), purity (P), and precision (Prec.), recall (Rec.), F1 on the fake (positive) class reported in % for the majority cluster. Metrics are reported for different clustering algorithms against different features (Feat.): (L)inguistics, (B)ehaviors. AMT data in Ott et al., (2011) does not have behavior information so values for B and L+B feature sets are nil. Improvements of LSM are significant (p<0.01, except entropy on the Yelp data which gives p<0.05) according to t-test over 50 runs.

$\boldsymbol{\xi}$). This resulted in 483 fake and 529 non-fake reviews.

Yelp Restaurants: This dataset is our own creation. We consider 2000 fake (filtered by Yelp) and 2000 non-fake (unfiltered) reviews by 601 reviewers from Yelp.com across 50 Boston restaurants. Yelp is a dedicated commercial review hosting site which has been performing industry scale filtering to remove fake or suspicious reviews since 2005 (Stoppelman, 2009). Recently there have been several works (Luca and Zervas, 2013; Feng et al., 2012a; Wang, 2010) which have used Yelp data for building deception models and studies that report Yelp filtering is reliable (Mukherjee et al., 2013). Hence, it should be safe to assume that filtered reviews (largely) correspond to deceptive fake reviews.

4.2 Systems

For comparison, we experiment with the following unsupervised clustering systems.

LSM + Uninformed Priors (LSM-UP): For this version of LSM, we set the Dirichlet prior as $\beta = 0.1$ as they seem to work well for language models (Griffiths and Steyvers, 2004). For Beta distributed variables, s^a, θ^f, $f \in \{EXT, DEV, ETF\}$, we set priors as follows. $\forall a \in A$, $\alpha^a \leftarrow (1, 1)$; $\gamma^f \leftarrow (1, 1)$. This uninformed setting makes any value in [0, 1] equally likely to be assigned to the Beta distributed variables. Priors for ψ^f, $f \in \{CS, MNR, ACT\}$ are estimated using the method-of-moments (Algorithm 1). Posterior estimates are drawn after 3000 iterations with an initial burn-in of 250 iterations.

LSM + Hyperparameter Estimation (LSM-HE): This setting estimates the hyperparameters α^a, $a \in A$ and γ^f, $f \in \{EXT, DEV, ETF\}$ using Algorithm 2 keeping all other settings same as LSM-UP.

Both LSM-UP and LSM-HE are generative models which produce a class/category assignment to each document (review). This is used to generate two clusters by placing each review (document) in the cluster of the assigned class/category.

Partitional and Hierarchical Clustering: We consider 3 clustering algorithms for our baseline. K-means is the most obvious unsupervised partitional algorithm for clustering spam (fake) and non-spam (non-fake) reviews. We also experiment with single and complete link agglomerative Hierarchical Clustering (HC). Lingpipe implementations of K-means and HC[3] were used. We tried both Euclidean and Cosine distance metrics but only report cosine results as they produced better results. Cosine has been shown to be better than Euclidean for text clustering in general (Huang, 2008).

4.3 Evaluation Metric

Clustering performance is usually evaluated using

[3] LingPipe produces a dendogram upon inducing a hierarchical clustering. Dendograms are modeled as binary trees which are linked with other parent and leaf (single element cluster) nodes. The method partitionK(int K) returns a partition of the data into K categories by breaking links in the order of decreasing distance until the specified number of partitions are generated.

purity and entropy (Manning et al., 2008). To compute purity, each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned instances and dividing by the total number of instances, N.

$$purity(\Omega, C) = \frac{1}{N} \sum_k \left(\max_j (\omega_k \cap c_j) \right) \tag{13}$$

where $\Omega = \{\omega_{k \in \{1 \ldots K\}}\}$ is the set of clusters and $C = \{c_{j \in \{1 \ldots J\}}\}$ is the set of classes, and $N = \sum_k |\omega_k|$. We interpret ω_k as the set of instances (reviews in our case) in cluster k and c_j as the set of instances (reviews) in class j. Bad clustering has purity close to 0 while a perfect clustering has purity 1. For binary clustering, purity is same as accuracy.

Our second metric is entropy. The entropy of a clustering reflects how the instances in the j (= 2) distinct classes (fake and non-fake) are distributed within each resulting cluster. Entropy is a global quality measure and is computed by averaging the entropy of all clusters as follows:

$$Entropy = -\sum_k \left(\left(\frac{|\omega_k|}{N} \right) \sum_{j=1}^{J} \left(\frac{|\omega_k \cap c_j|}{|\omega_k|} \log_2 \left(\frac{|\omega_k \cap c_j|}{|\omega_k|} \right) \right) \right) \tag{14}$$

In contrast to purity, entropy decreases as the quality of clustering improves.

Although purity and entropy are largely accepted metrics for clustering evaluation, they do not necessarily translate into good effectiveness in a particular application domain. Manning et al., (2008) recommends direct evaluation on the application of interest. In our case of fake review detection, this corresponds to precision, recall, and F1-score on the majority cluster (i.e., cluster containing more fake reviews). Higher F1 indicates that the algorithm is able to separate most of the fake reviews from non-fake reviews directly translating to better detection in an actual application. Thus, we report entropy, purity, precision, recall, and F1 for each system in Table 2. For K-means and HC, we further experiment with different features: (L)inguistic bag of words and (B)ehaviors.

4.4 Results

We note the following observations from Table 2:

1. Across Amazon and Yelp datasets, using only behavioral features (§3.2) give slightly better F1 for K-means and HC. Using linguistic and behaviors together further improve F1. Thus, both linguistics and behavioral features are useful.

Dropp-ed Feat.	E	P	Prec	Rec	F1	E	P	Prec	Rec	F1
CS	0.89	0.66	61.0	85.1	71.0	0.98	0.57	56.7	62.1	59.2
MNR	0.89	0.66	60.1	87.4	71.2	0.98	0.56	55.9	62.0	58.8
ACT	0.88	0.67	61.1	85.8	71.4	0.99	0.56	55.0	61.4	58.0
EXT	0.85	0.69	63.1	88.6	73.7	0.98	0.57	56.1	63.5	59.6
DEV	0.89	0.66	60.2	85.9	70.8	0.99	0.55	54.9	62.0	58.2
ETF	0.88	0.68	61.4	86.9	71.9	0.98	0.58	55.9	63.7	59.5

(a) Amazon (b) Yelp Restaurants

Table 3: Feature Ablation Experiments. Clustering performance after dropping each behavior feature from the full feature set (L+B using LSM-HE, last row in Table 2). Differences in metrics (from Table 2, last row) for each dropped feature are statistically significant with p<0.01 (except entropy on the Yelp data which is significant at *p*<0.05) based on paired *t*-test over 50 runs.

2. Using the full feature set (L+B), for the AMT data and the Yelp data, K-means performs better than complete-link HC. Single-link HC performs slightly poorer. For the Amazon data, complete-link HC performs better than K-means and single-link HC. On purity and entropy, the baselines (K-means, HC) perform somewhat similarly.
3. Purity and F1 of the generative models (LSM-UP, LSM-HE) are significantly (p<0.01, see Table 2 caption) better than K-means and HC. LSM-HE performs better than LSM-UP showing that hyperparameter estimation is useful. For the gold-standard AMT data and expert labeled Amazon data, LSM-HE obtains a respectable F1 of about 76% and 74%. This corresponds to about 12% improvement over the best competitor. LSM-HE also improves purity and F1 by 5-10% for the Yelp data. Results of all methods for the Yelp data are lower compared to AMT data and Amazon data. This may be because the Yelp dataset is harder for the unsupervised clustering task.
4. Across all datasets, the proposed generative models LSM-UP and LSM-HE significantly outperform baseline clustering methods on entropy, purity and F1. For AMT and Amazon datasets, LSM-HE obtains a respectable recall of 89%. Precision is not so high which is understandable as there is usually a tradeoff between precision and recall for unsupervised text clustering (Manning, et al., 2008).

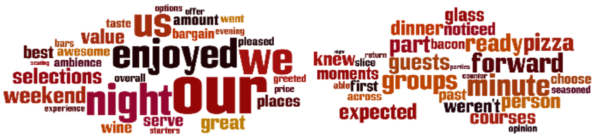
4.5 Feature Ablation Experiments

The previous results show that behavioral features are useful. It is naturally interesting to analyze individual feature contributions using ablation. We use the best performance setting LSM-HE (Table 2, last row) and drop each behavior feature from the full feature set L + B. AMT dataset is not used here as it does not contain behavior information. From Table 3, we see that dropping feature results in a graceful degradation in performance which shows that all behaviors are useful. Dropping CS, MNR, ACT, DEV features impacts clustering performance showing that they are more discriminating.

4.6 Linguistic Traces of Deception on the Web

In this section, we investigate the language models of spam/non-spam (fake/non-fake) reviews, i.e., the estimated posterior on $\varphi_{k\in\{\hat{s},\hat{n}\}}$ (see Table 1). Figure 2 shows the word clouds[4] for the Yelp restaurant dataset. We find that the fake reviews show more use of personal pronouns (e.g., *us*, *our*, *we*, etc.) and associated actions (e.g., *enjoyed*, *pleased*) towards targets (*night*, *weekend*, *wine*, etc.) with the objective of incorrect projection (lying/faking) which often involves more use of positive sentiments/emotions (e.g., *bargain*, *value*, *enjoyed*, etc.). Non fake reviews on the other hand show more balanced/natural distribution of words (e.g., *dinner*, *glass*, *groups*, *expected*, *moments*, etc.).

Studies on psycholinguistic deception (e.g., Newman et al., 2003) however state that lying/deceptive communications usually have fewer personal/first person pronouns. It is worthwhile here to understand the difference. Writing fake opinions/reviews on the Web is a distinctive cognitive/psychological process and not the same as conventional lying. Traditional lying/deceptive communications refers to statements of untrue facts (Newman et al., 2003). It involves the psychological process of "detachment" resulting in the use of fewer first-person pronouns. This phenomenon has been attested by researchers (e.g., Knapp et al., 1974; Buller et al., 1996) that liars tend to avoid statements of ownership to "dissociate" themselves resulting in fewer usage of first-person/personal pronouns. Fake reviews/opinions on the Web differ from conventional lies in two keys aspects. First, fake reviewers actually like to use more first-person pronouns such as *I*, *myself*, *mine*, *we*, *us*, etc., to make their reviews sound more convincing and to give readers the impression that their reviews express their "own" true experiences. We call this "attachment" as opposed to "detachment". Second, fake reviews may not be traditional lies of facts. For instance, an author of a book can pretend to be a reader of the book and write a review, or fake reviewers reviewing a product they never used, etc.



(a) Fake (b) Non fake

Figure 2: Word clouds in Yelp's Restaurant dataset

Thus, we see that deceptive opinion spam on the Web has subtle differences and complexities than traditional lying or deception as studied in the psycholinguistic literature. Fake review detection is thus a challenging problem. Our proposed models show promising results on multiple domains/datasets considering that our approach is unsupervised. Additionally, if richer internal/private data from websites (e.g., IP addresses, geo-location, session/network/click logs, mouse gestures, etc.) are available, more behaviors can be modeled which can significantly improve the detection accuracy of our approach. Further, our approach is very generic and can be applied to any review site for fake review detection as it relies only on review content, posting dates, ratings, etc. which are always available.

5 Conclusions

This paper proposed a novel way to utilize linguistic and behavioral clues to detect deceptive opinion spam (fake reviews) in an unsupervised Bayesian inference framework. The proposed model (LSM) treats opinion spam detection as a clustering problem. Learning exploits distributional divergence on linguistic and behavioral dimensions between spammers (fake reviewers) and other (non-spammers). The fully Bayesian approach facilitates modeling spamicity of authors and reviews as latent variables precluding the need of any labeled data. To the best of our knowledge, LSM is the first such model. To evaluate LSM, we conducted a comprehensive set of experiments across three opinion spam labeled datasets for deceptive opinion spam. The results showed that the proposed model significantly outperformed the baselines across all datasets. LSM's estimated language models also reveal interesting insights about the subtle linguistic traces left behind by spammers writing fake reviews and the linguistic process of deception on the Web.

[4] Created using Wordle with word sizes reflecting probabilities.

References

Bishop, C. M. 2006. Pattern recognition and machine learning. *Springer*. 2006.

Buller, D. B., Burgoon, J. K., Buslig, A., and Roiger, J. 1996. Testing Interpersonal Deception Theory: The language of interpersonal deception. *Communication Theory, 6, 268-289*.

Celeux, G., Chaveau, D., & Diebolt, J. 1996. Stochastic versions of the EM algorithm: An experimental study in the mixture case. *J. of Statistical Computation and Simulation, 55, 287{314}*.

Duda, R. O., Hart, P. E., and Stork, D. J. *Pattern Recognition*. *Wiley*. 2001.

Danescu-Niculescu-Mizil, C., Kossinets, G., Kleinberg, J., and L. Lee. 2009. How opinions are received by online communities: a case study on amazon.com helpfulness votes. *Proceedings of the international conference on World Wide Web (WWW)*.

Fei, G., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., & Ghosh, R. (2013, June). Exploiting Burstiness in Reviews for Review Spammer Detection. *Proceedings of the international AAAI Conference on Weblogs and Social Media (ICWSM)*.

Feng, S., Banerjee R., Choi, Y. 2012a. Syntactic Stylometry for Deception Detection. *Association for Computational Linguistics (ACL)*.

Feng, S., Xing, L., Gogar, A., and Choi, Y. Distributional Footprints of Deceptive Product Reviews. 2012b. *Proceedings of AAAI International Conference of Web and Social Media (ICWSM)*.

Griffiths, Thomas L., and Mark Steyvers. 2004. Finding scientific topics.*Proceedings of the National Academy of Sciences of the United States of America 101.Suppl 1 (2004): 5228-5235*.

Hancock,J.T., Curry,L.E., Goorha,S., and Woodworth,M. 2008. On lying and being lied to: A linguistic analysis of deception in computer-mediated communication. *Discourse Processes, 45(1):1–23*.

Huang, A. Similarity measures for text document clustering. 2008. *Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand*. 2008.

Jin,X.,Lin,C.X.,Luo,J.,Han,J. 2011. SocialSpamGuard: A Data Mining-Based Spam Detection System for Social Media Networks. *Proceedings of the Very Large Data Bases 4(12): 1458-1461*.

Jindal, N., Liu, B. and Lim, E. P. 2010. Finding Unusual Review Patterns Using Unexpected Rules. *ACM Conference on Information and Knowledge Management (CIKM)*.

Jindal, N., and Liu, B. Opinion spam and analysis. 2008. *Proceedings of the International Conference on Web search and web data mining (WSDM)*.

Kim, S.M., Pantel, P., Chklovski, T. and Pennacchiotti, M. 2006. Automatically assessing review helpfulness. *Empirical Methods in Natural Language Processing (EMNLP)*.

Knapp, M. L., Hart, R. P., & Dennis, H. S. 1974. An exploration of deception as a communication construct. *Human Communication Research, 1, 15-29*

Kolari, P., Java, A., Finin, T., Oates, T., Joshi, A. 2006. Detecting Spam Blogs: A Machine Learning Approach. *Proceedings of Association for the Advancement in Artificial Intelligence (AAAI)*.

Koutrika, G., Effendi, F. A., Gyöngyi, Z., Heymann, P., and H. Garcia-Molina. 2007. Combating spam in tagging systems. *Adversarial Information Retrieval and the Web (AIRWeb)*.

Lauw,H.W.,Lim,E.P.,Wang,K. 2006. Bias and Controversy: Beyond the Statistical Deviation. *ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*.

Lauw, H.W., Lim, E.P., Wang,K. 2007. Summarizing Review Scores of Unequal Reviewers. *In Proceedings of the SIAM conference in Data Mining (SDM)*.

Lee, H. and A. Y. Ng. 2005. Spam Deobfuscation Using a Hidden Markov Model. *In Proceedings of the Conference on Email and Anti-Spam (CEAS)*.

Li, F., Huang, M., Yang, Y. and Zhu, X. 2011. Learning to identify review Spam. *In Proceedings of International Joint Conference of Artificial Intelligence (IJCAI)*.

Lim, E. P., Nguyen, V. A., Jindal, N., Liu, B., & Lauw, H. W. 2010. Detecting product review spammers using rating behaviors. *Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM)*.

Liu, J., Cao, Y., Lin, C., Huang, Zhou, M. 2007. Low-quality product review detection in opinion summarization. *Empirical Methods in Natural Language Processing (EMNLP)*.

Luca, M., & Zervas, G. 2013. Fake It Till You Make It: Reputation, Competition, and Yelp Review Fraud. *Harvard Business School NOM Unit Working Paper*, (14-006).

Manning, C. D., Prabhakar R., and Hinrich S. 2008. Introduction to information retrieval. *Vol. 1. Cambridge: Cambridge University Press, 2008*.

Mihalcea, R. and Strapparava, C. 2009. The lie detector: Explorations in the automatic recognition of deceptive language. *Proceedings of ACL-IJCNLP (short paper)*.

Mukherjee, A., Kumar, A., Liu, B., Wang, J., Hsu, M., Castellanos, M., & Ghosh, R. 2013. Spotting opinion spammers using behavioral footprints. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*.

Mukherjee, A., Liu, B. and Glance, N. Spotting Fake reviewer groups in consumer reviews. *Proceedings of the international conference on World Wide Web (WWW)*.

Mukherjee, A., Venkataraman, V., Liu, B., & Glance, N. 2013. What Yelp Fake Review Filter might be Doing. In *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media (ICWSM)*.

Newman, L., Pennebaker,J.W.,Berry,D.S.,Richards,J.M. 2003. Lying words: predicting deception from linguistic styles, *Personality and Social Psychology Bulletin.*

Ott, M., Cardie, C., & Hancock, J. T. 2013. Negative Deceptive Opinion Spam. *In Proceedings of NAACL-HLT (pp. 497-501).*

Ott, M., Cardie, C. and Hancock, J. 2012. Estimating the prevalence of deception in online review communities. *Proceedings of the 21st international conference on World Wide Web(WWW)*..

Ott, M., Choi, Y., Cardie, C. Hancock, J. 2011. Finding Deceptive Opinion Spam by Any Stretch of the Imagination. *Association of Computational Linguistics (ACL)*. 2011.

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. 1998. A Bayesian approach to filtering junk e-mail. In *Learning for Text Categorization: Papers from the 1998 workshop* .

Serrano, M. A., Flammini, A. and Menczer, F. 2009. Modeling statistical properties of written text. *PloS one, 4(4):5372*.

Smyth, P. 1999. Probabilistic Model-Based Clustering of Multivariate and Sequential Data. In *In Proceedings of Artificial Intelligence and Statistics (AISTATS)*.

Spirin, Nikita, and Jiawei Han. 2012. Survey on web spam detection: principles and algorithms. *ACM SIGKDD Explorations Newsletter* (2012): 50-64.

Stoppelman, J. 2009. Why Yelp has a Review Filter. http://officialblog.yelp.com/2009/10/why-yelp-has-a-review-filter.html. Yelp Official Blog. October 2009.

Streitfeld, D. 2012. Fake Reviews Real Problem. New York Times. http://query.nytimes.com/gst/fullpage.html?res=9903E6DA1E3CF933A2575AC0A9649D8B63

Wang, G., Xie, S., Liu, B., and Yu, P. S. 2011. Review Graph based Online Store Review Spammer Detection. *International Conference on Data Mining (ICDM)*.

Wang, Z. 2010. Anonymity, social image, and the competition for volunteers: a case study of the online market for reviews. *The BE Journal of Economic Analysis & Policy*.

Wu, G., D. Greene, B. Smyth, and P. Cunningham. 2010. Distortion as a validation criterion in the identification of suspicious reviews. *Technical report, UCD-CSI- 2010-04, University College Dublin*.

Xie, S., Wang, G., Lin, S., and Yu, P. S. 2012. Review Spam Detection via Time Series Pattern Discovery, *Proceedings of the ACM conference on Knowledge Discovery and Data Mining (KDD)*.

Yu, F, Xie, Y., and Ke, Q. 2010. Sbotminer: large scale search bot detection. *Proceedings of the third ACM international conference on Web search and data mining (WSDM)*.

Zhou, L., Shi,Y., and Zhang,D. 2008. A Statistical Language Modeling Approach to Online Deception Detection. IEEE Transactions on Knowledge and Data Engineering.

Zhu, C., Byrd, R. H., Lu, P., & Nocedal, J. 1997. Algorithm 778: L-BFGS-B: Fortran routines for large scale bound constrained optimization. *ACM Transactions on Mathematical Software, 23, 550{560}*.