Uncovering Crowdsourced Manipulation of Online Reviews

Amir Fayazi
Texas A&M University
College Station, TX 77843
afayazi@tamu.edu

Kyumin Lee
Utah State University
Logan, UT 84322
kyumin.lee@usu.edu

James Caverlee
Texas A&M University
College Station, TX 77843
caverlee@cse.tamu.edu

Anna Squicciarini
Pennsylvania State University
University Park, PA 16801
asquicciarini@ist.psu.edu

ABSTRACT

Online reviews are a cornerstone of consumer decision making. However, their authenticity and quality has proven hard to control, especially as polluters target these reviews toward promoting products or in degrading competitors. In a troubling direction, the widespread growth of crowdsourcing platforms like Mechanical Turk has created a large-scale, potentially difficult-to-detect workforce of malicious review writers. Hence, this paper tackles the challenge of uncovering crowdsourced manipulation of online reviews through a three-part effort: (i) First, we propose a novel sampling method for identifying products that have been targeted for manipulation and a seed set of deceptive reviewers who have been enlisted through crowdsourcing platforms. (ii) Second, we augment this base set of deceptive reviewers through a reviewer-reviewer graph clustering approach based on a Markov Random Field where we define individual potentials (of single reviewers) and pair potentials (between two reviewers). (iii) Finally, we embed the results of this probabilistic model into a classification framework for detecting crowd-manipulated reviews. We find that the proposed approach achieves up to 0.96 AUC, outperforming both traditional detection methods and a SimRank-based alternative clustering approach.

Categories and Subject Descriptors: H.3.5 [Online Information Services]: Web-based services

Keywords: crowdsourced manipulation; deceptive review; amazon; crowdsourcing site; review site

1. INTRODUCTION

Reviews are a ubiquitous component of online commerce – from hotel and travel booking sites (e.g., Expedia, Trip Advisor and hotels.com) to e-commerce sites (e.g., Amazon and eBay) to app stores (e.g., Google Play and Apple's App Store). Online reviews provide a voice for customers to praise or criticize a product or service, often in conjunction with a star rating, providing helpful information to future potential customers. Unsurprisingly, research has shown that product review ratings are, indeed, correlated with sales [4, 10].

SIGIR'15, August 09 - 13, 2015, Santiago, Chile.

DOI: http://dx.doi.org/10.1145/2766462.2767742.

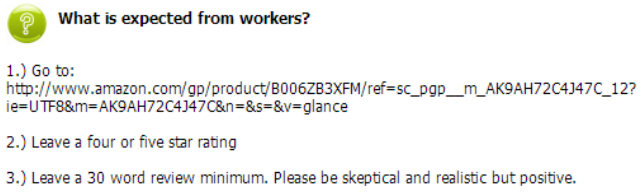

Figure 1: An example deceptive review task posted to a crowdsourcing platform.

Naturally, there is an incentive to pollute these reviews toward promoting one's products or in degrading one's competitors. This pollution has been identified as a growing threat to the trustworthiness of online reviews by major media and by the research literature [7, 20, 22, 25, 26]. Recently, Ott et al. suggested that up to 6% of reviews on sites like Yelp and TripAdvisor may be deceptive [22]. These reviews can have serious consequences: a deceptively promoted weight loss supplement identified as part of this project led one customer to write: *... these pills made me really sick, palpitations, increased heart rate and troubled breathing. I requested a return, and the response I got from the company selling these horrible pills was very rude....*

In a troubling direction, the widespread growth of crowdsourcing platforms has created a new attack vector for polluting online reviews. Crowdsourcing platforms like Amazon Mechanical Turk have been hailed for their effectiveness in intelligently organizing large numbers of people [3, 11]. These platforms however also enable a large-scale, potentially difficult-to-detect workforce of deceptive review writers [17, 28]. To illustrate, Figure 1 shows an example task that asks each worker to leave a high product rating and a "skeptical and realistic but positive" review. Figure 2 shows a sample of two deceptive reviews written in response to this type of crowdsourced task. Compared to traditional spam bots that typically leave identifiable footprints [15, 16, 18], these human-powered deceptive reviews are inherently distinct, linked only by their common theme and not in common keywords, phrases, or other easily identifiable signals. And since crowdsourced deceptive reviews are generated by humans rather than bots, their ongoing detection is even

★★★★★ This pills are simply the best ! I tried so many times to lose some weight, but this is the only thing that really worked for me ! You have to try it, this is money well spend .

★★★★★ Slimula works. I must say that slimula was very strong for the first two days but then my body got use to it and I'm losing weight

Figure 2: Two crowdsourced reviews for a weight-loss product sold by Amazon.

more challenging since crowds can actively circumvent detection methods.

Unfortunately, there is a significant gap in our understanding of crowdsourced manipulation of online reviews and effective methods for uncovering such manipulation, due to a number of challenges. One critical challenge is a lack of clear ground truth for analyzing deceptive reviews and in building countermeasures; it is difficult to ascertain which reviews are deceptive and which ones are legitimate. A second important challenge is that polluters may rely on multiple communication channels to coordinate their activities – including private methods such as email and instant messenger which are difficult to observe – and so deceptive intent may be further obscured. To tackle these issues, we propose a three-part effort:

- First, we propose a novel sampling method for identifying products that have been targeted for manipulation and a seed set of deceptive reviewers who have been enlisted through the command-and-control of crowdsourcing platforms. To our knowledge, this is the first effort toward "pulling back the curtain" to uncover clear evidence linking deceptive intent with actual reviews.
- Second, we augment the seed set of sampled deceptive reviewers to identify additional deceptive reviewers who participate as part of hidden groups of coordinated manipulation. To capture the hidden infrastructure underlying deceptive reviews, we exploit connections between reviewers through a reviewer-reviewer graph clustering approach based on a conditional random field that models individual potentials (of single reviewers) in combination with pair potentials (between two reviewers).
- Finally, we embed the results of this probabilistic model into a classification framework for detecting crowd manipulated reviews. We find that the proposed approach achieves up to 0.96 AUC, outperforming both traditional detection methods and an alternative SimRank-based clustering approach.

2. RELATED WORK

In this section, we summarize previous research work related to deceptive reviews and crowdsourced manipulation.

First of all, in the deceptive or spam review research field, researchers have analyzed the growth of deceptive reviews over time and have studied how to detect these reviews. For example, Ott el al. [22] have reported that the number of deceptive reviews has grown across multiple consumer-oriented review sites. Danescu-Niculescu-Mizil et al. [5] suggested an underlying model for measuring review quality. They reported that when controversy of a product was high, some of the reviews whose ratings were different from the average rating of the product often got higher helpfulness scores. Other researchers discovered distinguishing patterns between spam and legitimate reviews from product review ratings [9] and temporal distribution of spam reviews [29].

A related line of research has studied coordinated groups of review spammers. Lu et al. [19] found that the social context of reviews was useful to find groups of review spammers. They showed that one can assess the quality of reviews with higher accuracy when one assumed the quality of a reviewer depended on the quality of her peers in the social network. Mukherjee et al. [21] found that analyzing groups of review spammers revealed clearer evidence of spam reviews than analyzing an individual spam review. Ott et al. [23] showed that linguistic features were not reliable for humans to distinguish between deceptive reviews and legitimate reviews. The proposed sampling method, reviewer clustering, and deceptive review classifier in this paper complement these existing approaches.

Next, researchers have begun to study the crowdsourced manipulation problem of spreading manipulated contents to target sites such as social networking sites, review sites, and search engines [17, 27, 28]. Wang et al. [28] analyzed the use of two Chinese crowdsourcing platforms, and estimated that 90% of all tasks were malicious tasks. Lee et al. [17] analyzed tasks in Western crowdsourcing platforms, and found that the primary targeted systems were online social networks including review sites (56%) and search engines (33%). But, the previous research work did not specifically focus on a crowdsourced deceptive review problem. Researchers expressed that labeling fake reviews by human is sometimes hard [13]. Our research is the first work to link crowdsourced deceptive review tasks to target products, and analyze these deceptive reviews and behaviors of deceptive reviewers toward building a deceptive review classifier.

3. OVERALL FRAMEWORK

Our ultimate goal is to identify deceptive online reviews. We formulate this problem as a classification problem, where the goal is to assign a class label of *deceptive* or *legitimate* to a candidate review r based on a classifier c:

$$c : r \rightarrow \{deceptive, legitimate\}$$

The first critical challenge to building an effective classifier is in identifying valid ground truth training data. Typical methods include: (i) rule-based heuristics – like labeling as deceptive all reviews containing a particular keyword or some other intuitive signal – however, such methods are brittle to changes in the strategies of deceptive reviewers; and (ii) asking human labelers to assess reviews, however these labelers typically do not know the intent of the original review writer, and so the labels may be in error. In contrast, we propose instead to sample directly from crowdsourcing platforms that publicly advertise review tasks.

3.1 Sampling Crowdsourcing Platforms for Deceptive Reviews

Our sampling strategy is first to collect tasks from crowdsourcing platforms that require workers to post a review on a target site (e.g., Amazon, Yelp) akin to the example task in Figure 1. By linking these tasks to products in a target site, we can then sample the products targeted, the reviews associated with the product, as well as the reviewers who contributed the reviews. The overall sampling framework is illustrated in Figure 3.

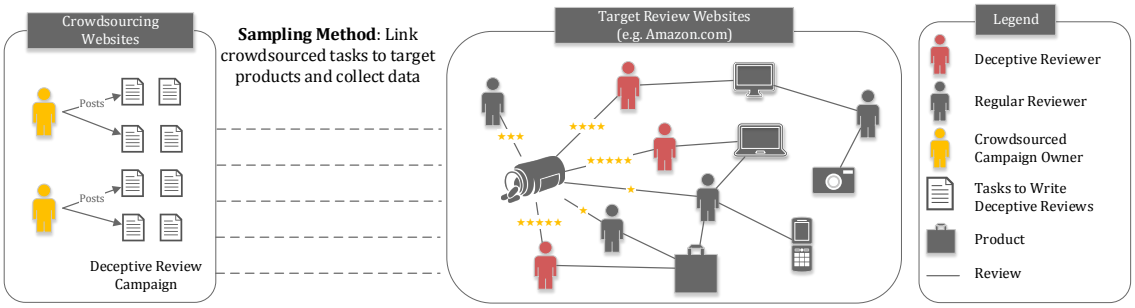

Figure 3: Overall Sampling Framework.

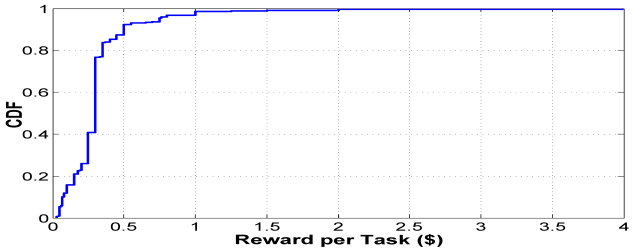

Figure 4: Cumulative Distribution Function of reward per task targeting Amazon.

In particular, we sampled tasks posted to three crowdsourcing platforms over a span of several weeks in 2013: RapidWorkers.com, ShortTask.com, and Microworkers.com. Each of these platforms supports a variety of tasks, typically offering on the order of $0.25 per task completion. We sampled all tasks on these platforms that target Amazon. Figure 4 presents a cumulative distribution function (CDF) of reward per task targeting Amazon. The average reward per task was $0.31, and the biggest reward per task was $3.99. There were various tasks targeting Amazon such as manipulating the number of clicks of a product page and the helpfulness of a review, downloading/purchasing a product and writing deceptive reviews. Among these tasks targeting Amazon, the majority tasks were to write deceptive reviews.

To study crowdsourced manipulation of reviews on Amazon, we extracted the Amazon URL from each task which links to a product page on Amazon, collected the product page, all reviews associated with the product, and all reviewers' information. In total, we identified 1,000 products on Amazon that had been the target of deceptive reviews. We call this the *root dataset*.

We augmented this base set of products, reviews, and reviewers via a breadth-first search crawling method. Specifically, we collected the previous Amazon reviews of each reviewer in the root dataset and the Amazon product pages those reviews were associated with. We subsequently collected all of the new reviews and new reviewer profiles from these product pages. In total, we crawled out three hops from the root dataset to identify a larger candidate set of potential deceptive reviews, reviewers, and targeted products. In total, this *expanded dataset* includes 71.1k reviews, 14.5k reviewers, and 48.9k products.

3.2 Clustering Deceptive Reviewers

Given the expanded dataset consisting of reviewers and products, our next goal is to reveal hidden linkages among the reviewers towards uncovering previously unknown deceptive reviewers who participate as part of hidden groups of coordinated manipulation. The original sampling method identifies products (in the root set) for which we are certain reviewers have targeted for manipulation. However, many efforts to pollute reviews may rely on private command and control (and so be out of reach of our sampling method) [28]. Can we exploit the linkages among reviewers and products to identify these hidden deceptive reviewers? This section presents a Reviewer-Reviewer graph clustering approach based on a pairwise Markov Random Field (MRF) that defines individual potentials (of single reviewers) and pair potentials (between two reviewers). In addition to providing evidence of coordinated manipulation, the output of this clustering approach can be integrated as an additional feature for building the deceptive review classifier.

3.2.1 *Building the Reviewer-Reviewer Graph*

Our root dataset indicates that the payoff for a single favorable review is generally less than a US dollar, so users are likely to write multiple reviews (about 5 each, in our dataset). As we assume that some reviewers cooperate with one another in targeting certain products, we build a reviewer-reviewer graph to capture these subtle dependencies, and detect "non-crowdsourcing" deviant reviewers.

Our expanded dataset can naturally be represented as a Reviewer-Product graph $G_{R,P}$. The Reviewer-Product graph is a bipartite graph of consumer products connecting reviewers with products, with edges representing reviews.

Using this information, we can construct the Reviewer-Reviewer graph $G_{R,R} := (R, E, W)$ where the vertex set R denotes the reviewers who wrote reviews on Amazon and the weighted edges (arcs) in E indicate how many pairs of reviewers have reviewed the same item during a short time window. Finally, W is the set of edge weights. Edge weights are calculated according to the weighting function $w : E \rightarrow W$, as follows.

$$w_{R,R}(a,b) := \frac{|\{p | p \in N_{R,P}(a) \wedge p \in N_{R,P}(b) \wedge |t(a,p) - t(b,p)| < T\}|}{\min\left(|N_{R,P}(a)|, |N_{R,P}(b)|\right)}$$

where $N_{R,P}(a)$ is the set of products rated favorably by a, $t(a,p)$ is the time when a reviewed p, and finally T is a constant time window size which was set to three days in our case. In essence, the numerator captures reviewer similarity by counting the number of products both a and b rated favorably in the same time window.

Following the projection of $G_{R,P}$ onto $G_{R,R}$ we prune edges whose weight denominator is 1.0 and then vertices with degree zero from $G_{R,R}$. The former is to avoid creating a strong ($w = 1.0$) collaboration edge between two reviewers due to a single review co-occurrence.

3.2.2 Modeling Reviewer Potentials

Given this Reviewer-Reviewer graph, we now describe a probabilistic approach for clustering reviewers (nodes) toward identifying groups of similar reviewers. We formulate this approach using a MRF defined over the Reviewer-Reviewer graph $G_{R,R}$. Each node in the random field corresponds to a reviewer and is a discrete random variable whose value is the cluster the corresponding reviewer belongs to.

The proposed approach captures the following intuition:

- If two reviewers have collaborated heavily, they should be assigned to the same cluster. This allows for a clustering where connectedness is the criterion, and clusters can expand in a non-convex fashion similar to a spatial clustering. This kind of clustering does not require a precisely pre-determined number of clusters. Instead, it requires assigning a reasonably large number of clusters, and the clustering process will eventually choose the best number of clusters by merging initial clusters (e.g., initially assign 50 clusters, but finally end up with four large clusters).
- Toward creating cohesive clusters, we additionally require that all the reviewers of a cluster display similar features. For instance, reviewers who have actually purchased a product they reviewed and reviewers who did not purchase the product they have reviewed should belong to different clusters.

We model reviewers using two potential functions, capturing the intuitions above. The first is a *singleton potential*, as it is defined per each reviewer ϕ_j. The other type of function is per pair of reviewers ϕ_{pair}, and hence called *pair potential*. The singleton potentials have higher value when the reviewer features are close to that of the mean cluster features in the feature space. The pair potentials are higher when two connected reviewers in the graph are assigned to the same cluster. We aim to determine the most likely cluster assignments given the Reviewer-Reviewer graph and these potential functions are used to factorize the probability distribution:

$$P(\boldsymbol{Z}|\boldsymbol{\theta}, \boldsymbol{D}) \propto \prod_j \phi_j(Z_j) \prod_{j,k} \phi_{\text{pair}}(Z_j, Z_k)$$

where D stands for observations from the data, and is bolded to signify a set of random variables; Z represents cluster assignments which is not observed (hidden data); θ represents the parameters of the model; $\phi_j(Z_j)$ is the potential function over the labeling of reviewer j and ϕ_{pair} is over pairs of labels. Since the MRF is wholly conditioned on the observation data D, it is an instance of a Conditional Random Field (CRF). Next we will describe the factors (potentials) in our model.

Singleton Potentials. The singleton potential function is designed to capture how well an individual labeled reviewer corresponds to the cluster in which it belongs. Formally, we define the singleton potential as:

$$\phi_j(Z_j) = \Pr(Z_j, \overbrace{\boldsymbol{F_j}, \boldsymbol{P_j}}^{\text{Observations}} |\boldsymbol{\theta}) = \Pr(\boldsymbol{F_j}, \boldsymbol{P_j}|Z_j, \boldsymbol{\theta}) \Pr(Z_j|\boldsymbol{\theta})$$

where the variables $\boldsymbol{F_j}$ and $\boldsymbol{P_j}$ are the observed data and Z_j is the hidden variable (cluster of reviewer j). $\boldsymbol{\theta}$ represents the model parameters. $\boldsymbol{F_j}$s are the features of a reviewer j. $\boldsymbol{P_j}$s are the products reviewed by reviewer j. In practice, we can adopt any of a number of features that may be good candidates for distinguishing between deceptive and legitimate reviewers. In this paper, we adopt the following six features $\boldsymbol{F_j}$s in the singleton potential function:

- Real Name: Does a reviewer have a real name verified by Amazon? – *Binary*
- Helpfulness of reviews: Are the majority of a reviewer's reviews helpful? – *Binary*
- Verified Purchase: The majority of a reviewer's reviews are based on a verified purchase – *Binary*
- Length of the reviews written – *Log normal*
- Are favorable reviews (4, 5 star) more helpful than other reviews (1 through 3 stars)? – *Binary*
- Whether reviewer has more verified purchases for favorable reviews – *Binary*

We additionally make the assumption that feature values are conditionally independent given the cluster (as illustrated in Figure 5, which shows a graphical model of our independence assumption). As a result, the conditional probability of the observed data is:

$$\Pr(\boldsymbol{F_j}, \boldsymbol{P_j}|\theta, Z_j) = \prod_{F_{jk} \in F_j} \Pr(F_{jk}|Z_j, \theta) \prod_{j \sim k} \Pr(P_{jk}|Z_j, \theta)$$

where the symbol $\sim$ stands for adjacency in the graph.

We can model the binary variables (e.g., *Real Name*, *Helpfulness*) as Bernoulli random variables:

$$\Pr(F = f|Z = c) = \begin{cases} p_c & f = 1 \\ 1 - p_c & f = 0 \end{cases}$$

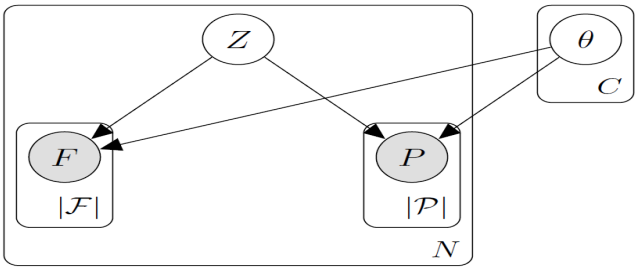

Figure 5: The independence assumption of the variables during calculation of singleton potentials ϕ. The indicator variables P correspond to each product and indicates whether that product was reviewed favorably. We assume the probability of that only depends on which cluster Z the author belongs to and the preference of that cluster θ. F represents various features of the authors in the cluster.

Since we find that the length of reviews has a log-normal distribution, the average log-length of reviews of a reviewer follows a Gaussian distribution which is denoted by L in the following:

$$\Pr(L = \log l | Z = c) = \mathcal{N}(\log l; \mu_c, \sigma)$$

Note that we do not require that each cluster of coordinated reviewers be equally likely to review all the products. Rather, we assume that each reviewer has concentrated on writing reviews for a subset of the products in the cluster. This is modeled by the second part of the singleton potential function. Given the cluster, there is a probability distribution over products π_c where product i has the chance $\pi_{c,i}$ of being reviewed by a reviewer in a cluster c. This part of the singleton potential tends to push dissimilar set of reviewers into separate clusters.

$$\Pr(P_i | Z_j = c) = \pi_{c,i}$$
$$\text{Subject to} \sum_i \pi_{c,i} = 1$$

Pair Potentials. The second potential function of the MRF likelihood formulation is the one between pairs of reviewers. The role of this function is to force that reviewers who have collaborated on writing favorable reviews, end up in the same cluster. In addition, if the number of clusters is larger than the actual underlying number of hidden groups, this potential function will regulate the resulting clustering where too many distinct clusters with edges between them are punished. The likelihood of two adjacent reviewers to be in the same cluster in the graph depends on a tunable parameter τ and on the weight of collaboration between the two reviewers $w_{R,R}(j,k)$.

$$\phi_{\text{pair}}(Z_j, Z_k) = \begin{cases} \tau^{w_{R,R}(j,k)} & Z_j = Z_k \\ (1-\tau)^{w_{R,R}(j,k)} & Z_j \neq Z_k \end{cases}$$

By increasing the value of τ, connected reviewers are more likely to be in the same cluster.

3.2.3 Learning Parameters and Clusters

Based on the probabilistic modeling part described in the previous section, we now have the following:

- Model parameters θ: including μ_c, p_cs, and $\pi_{c,i}$
- Hidden data Z_j: cluster assignments
- Observations: including F_k, P_m, and $w_{R,R}$

We aim to maximize the likelihood $\mathcal{L}(\boldsymbol{Z}, \boldsymbol{\theta}; D)$. That is, we need to see what sort of cluster parameters θ and cluster membership Z fit our evidence (i.e., our reviewer-reviewer graph) D the most. A common way to deal with maximizing likelihood functions which depend on hidden data (Z) is to use a local optimization method like Expectation Maximization (EM). EM is a popular method for missing data problems and has been successfully applied to similar MRF models in other domains [2]. EM iterates over two steps to increase the likelihood [6]. However it might get trapped in a local maximum. Therefore, we use a variant of EM, called *hard EM* with random restarts, which consists of the following steps:

1. Initialize parameters θ^0 to random values
2. While parameters have not converged:

   E-Step Calculate maximum a posteriori (MAP) values for $Z^{(t)}$ given $\theta^{(t)}$. That is, find the best cluster assignment given current parameters

   M-Step Calculate maximum likelihood estimate (MLE) $\theta^{(t+1)}$ given the $Z^{(t)}$. That is, find better parameter estimates given current cluster assignments

In the following, we present our parameter initialization and EM steps.

Parameter Initialization. The p_c parameters for binary features are sampled from a Dirichlet distribution $\text{Dir}([\alpha, \alpha])$. α in this case is a hyper-parameter for the p_c parameters. We pick α values relatively large so the sampled p_c's don't end up being close to the extremes (0 or 1). Instead they will be (in this case) close to 0.5 with a little perturbation. The initialization distribution for $\pi_{c,i}$ is similarly $\text{Dir}([\alpha_1, \cdots, \alpha_P])$ where P is the number of all products and α_is have the same value. For large α the resulting $\pi_{c,i}$ is close to uniform distribution with a little perturbation which is what we desired. The μ_cs of review lengths are initialized uniformly in the range of $[0, \max\{\log(\text{review length})\}]$. The parameter σ for review lengths is fixed at 1.

E-Step. In this step we should assign cluster labels so the following log likelihood function is maximized.

$$\log \mathcal{L}(Z, \boldsymbol{\theta}) = \sum_j \log \phi_j(Z_j) + \sum_{j,k} \log \phi_{\text{pair}}(Z_j, Z_k)$$

We adopt an Integer Program formulation of this problem by Kleinberg and Tardos called *Uniform Metric Labeling* [14]. For each node j (reviewer in the graph) we define an indicator variable $x_{j,c}$. If $x_{j,c} = 1$, it indicates node j is assigned to cluster c. In the pair potentials part of the summation, for each edge (j,k) the variable $d_{jk} = \frac{1}{2}\sum_{c \in C} |x_{j,c} - x_{k,c}|$ is the binary distance between the assigned clusters of j and k, where 0 means identical clusters and 1 is different clusters. For each edge of the reviewer-reviewer graph, we then have the potential:

$\log \phi_{\text{pair}}(Z_j, Z_k) = w_{R,R}(j,k)\,(d_{jk} \log(1-\tau) + (1-d_{jk}) \log \tau)$.

The Integer Program formulation can be relaxed to a Linear Program:

Minimize

$$\sum_{j \in R, c \in C} -\log \phi_j(Z_j = c) x_{jc} + \sum_{(j,k) \in E_{R,R}} -w_{R,R} \left(\log \frac{1-\tau}{\tau} \right) d_{jk}$$

Subject to

$$\sum_{c \in C} x_{j,c} = 1$$
$$d_{jk} = \frac{1}{2} \sum_{c \in C} d_{jkc}$$
$$d_{jkc} \geq x_{jc} - x_{kc}$$
$$d_{jkc} \geq x_{kc} - x_{jc}$$
$$x_{kc} \geq 0$$

Once the optimum x_{jc} are calculated, we pick the c with the highest x_{jc} as the cluster assignment for reviewer j.

M-Step. In the M-Step we update model parameters θ with their maximum likelihood estimate (MLE) given the cluster

assignments. The MLE estimates can be simply determined using frequency counts.

$$\Pr(Z = c) = \frac{|\{j \in R \mid Z_j = c\}|}{|R|}$$

$$\pi_{ci} = \frac{|\{(j,i) \in E_(R,P) \mid j \in R, i \in P, Z_j = c\}|}{S}$$

In the denominator, S is the normalizing factor so $\sum_{i \in P} \pi_{ci} = 1$. Similarly, the value of μ_c is updated as follows.

$$\mu_c = \frac{\sum_{a \in C} l_a}{|c|}$$

Similarly, the values for p_cs for various binary features of clusters can be determined with frequency counts.

3.3 Summary

To summarize, our goal toward identifying deceptive reviews (and reviewers) is based on the following logic. First, we sample known products that have been the target of deceptive reviews by crawling several crowdsourcing platforms. We call this the *root dataset*. We augment this base set of products, reviews, and reviewers via a breadth-first search crawling method to identify the *expanded dataset*. Given this expanded dataset, we aim to reveal hidden linkages among reviewers via a probabilistic clustering approach. The output of this clustering are groups of reviewers who often posted reviews to same products. In addition to providing additional evidence of coordinated manipulation, the output of this clustering approach can be integrated as an additional feature for building the deceptive review classifier.

4. EXPERIMENTS

In this section, we evaluate two main sets of experiments. First, we evaluate the reviewer clustering method over: (i) a synthetic dataset to validate the method's capacity to uncover groups of coordinating reviewers; and (ii) the expanded dataset to determine if meaningful groups of deceptive reviewers can be discovered from a real Amazon dataset. Second, we build a deceptive review classifier by using the output of the clustering algorithm as an additional feature and evaluate its performance versus a baseline without knowledge of the clustering output. We then compare this cluster-aware deceptive review classifier versus one based on an alternate clustering method (SimRank + k-medoids) to further validate the design choices in our reviewer-reviewer clustering method.

4.1 Identifying Reviewer Clusters

Our first goal is to understand how well the proposed clustering method performs. This is an important step for connecting known deceptive reviewers to potentially unknown ones. We first consider a synthetic dataset designed to contain natural groupings; does the proposed clustering method recover these underlying groups? Based on these observations, we then apply the clustering method over the *expanded dataset* based on 71.1k reviews, 14.5k reviewers, and 48.9k products. Do we discover groups engaged in coordinated manipulation?

4.1.1 With Synthetic Data

We begin by considering a synthetic dataset that is generated according to the same process the model in Section 3.2 assumes as detailed in Algorithm 1. The data generation procedure takes as input the number of clusters (K) and a total number of reviewers (N), and outputs a synthetic collaboration graph of reviewers. The relative sizes of each of K clusters is sampled from a Dirichlet distribution which is commonly used as a prior distribution for a multinomial. The parameter α determines how variant/uniform the clusters sizes will be. This allows for generating various proportions of cluster sizes. Then for each cluster, all its parameters are sampled.

Algorithm 1 Synthetic data generation procedure

Input: K (no. clusters), N (no. reviewers)
Input: Dirichlet params $(\alpha_1, ..., \alpha_K)$, τ
▷ Sample cluster parameters
Sample cluster sizes $\sim$ Dir$(\alpha, ..., \alpha)$
for each cluster c **do**
Sample μ_c (mean) of review log-length $\sim$ Gaussian
Sample $p_{c,\text{feature}}$ for features {Helpfulness, RealName, VerifiedPurchase} $\sim$ Uniform $0-1$
Sample cluster product preferences $\pi_c \sim$ Dir$(\beta, ..., \beta)$
end for
▷ Sample reviewers' features and edges between them
Assign reviewers to clusters
for each cluster c in K clusters **do**
for each reviewer n in cluster c **do**
Sample reviewer features RealName, Helpfulness, VerifiedPurchase $\sim$ Bernoulli$(p_{c,\text{feature}})$
end for
end for
for each reviewer a, b pairs **do**
if cluster(a) = cluster(b) **then**
add an edge if Bernoulli$(\tau) = 1$
else
add an edge if Bernoulli$(1-\tau) = 1$
end if
end for

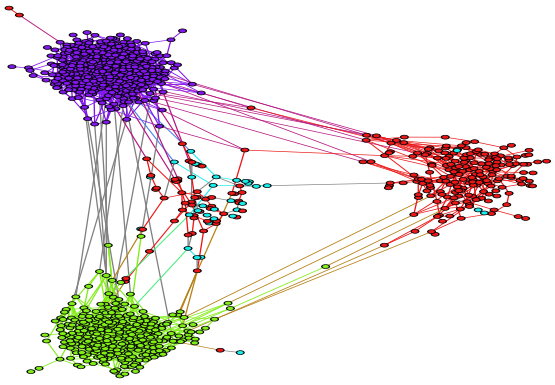

Figure 6: The proposed clustering method identifies four clusters from a synthetic graph consisting of 950 nodes and 2,150 edges. Members of each cluster are colored differently.

Once all cluster parameters are sampled, for each cluster, $|C_i|$ nodes are generated where $|C_i|$ is the cluster size of cluster i. The node values are sampled from the cluster parameter as illustrated in Figure 5. Finally, the edges are sampled. The average degree is kept constant (30 in our

Table 1: Rand Index results of our reviewer clustering approach on synthetic data.

No. Clusters	4	6	10	15
Avg. Rand Index	0.87	0.85	0.85	0.91

case). So for each possible edge, if it is between two nodes of the same cluster, it occurs with probability τ and if the edge is between two dissimilar clusters, it occurs with probability $1-\tau$. An example resulting graph of such a process with 4 clusters is shown in Figure 6. We run our method by over-estimating the number of clusters as 10 and use the same τ that was used by the generation process. Using higher values for τ mostly resulted in two of the detected clusters being labeled as the same cluster.

As shown in Figure 6, our clustering method successfully recovered most of the clusters as long as the intra-cluster edges occurred more frequently than inter cluster edges. Most mistakes happened in the central cluster where the density was low. The clustering method recovered 4 clusters which matched the true number of clusters, and 94.3% of the nodes were clustered correctly.

As another evaluation metric, we used *Rand Index*, a well-known metric for evaluating the quality of clustering when the ground truth is known. Graphs of size 1.2K nodes and 2.6K edges with 4 clusters were generated using the process described earlier. Given different numbers of predetermined clusters to the clustering algorithm, we list the average Rand Index of 10 runs in Table 1. One noticeable point in Table 1 is the improved clustering performance when the number of predetermined clusters over-estimates the actual number. The reason is that the clustering method is based on EM which is a local optimizer of the likelihood. More clusters with random initial parameters spread out in the parameter space mean a better chance of finding a more optimum final likelihood, so we observe better clustering results.

These results suggest that the proposed clustering method works well; but does it uncover meaningful groups over the expanded dataset?

4.1.2 With Real Data

Based on the expanded dataset of products, reviews, and reviewers, we formed the Reviewer-Reviewer graph as described in Section 3.2.1. In the graph, nodes are reviewers, and edges appear when a pair of reviewers write favorable reviews for the same products in a short time frame. The higher number of same products a pair of reviewers posts reviews to, the larger the edge weight of the reviewers is. This graph is shown in Figure 7. As a cleaning step, we discarded small connected components of this graph; we kept reviewers who belonged to a connected component of size 10 or more. We modeled reviewers using the six features described in Singleton Potentials in Section 3.2.2.

Next, we applied the described clustering method on the resulting reviewer graph. The EM algorithm was run with 16 random restarts. The predetermined number of clusters was set to 10. The value of τ determines how likely connected nodes belong to the same cluster. One of the strengths of this clustering method is that connected nodes with dissimilar clusters are punished regardless of the number of predetermined clusters. Hence, the eventual number of emerged clusters can be less than what is predetermined. For a high value of τ like 0.99 we ended up with almost one cluster for all the nodes. For lower value of $\tau = 0.7$ we ended up with 6 clusters, 3 of which had higher densities. Since we got a reasonable result returning clusters with high densities when we set $\tau = 0.7$, we used this value for experiments.

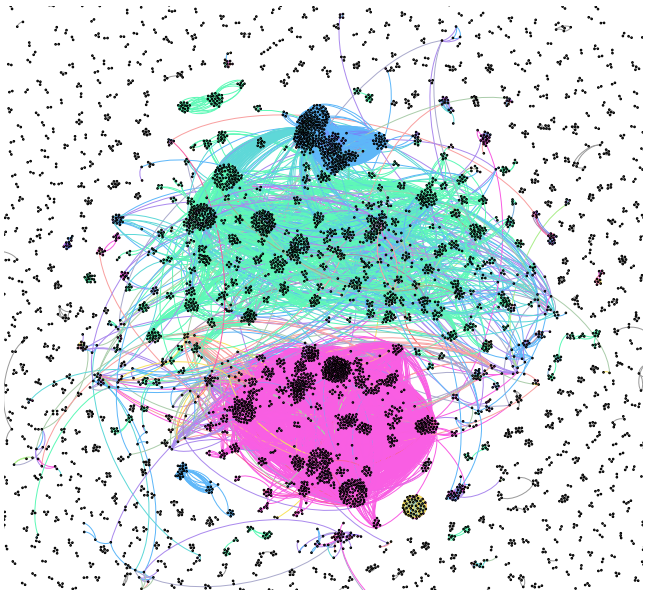

Figure 7: Detected clusters in the Amazon reviewer-reviewer graph. Nodes and edges are colored based on the cluster to which they were assigned. Two out of the three largest clusters are deceptive reviewer clusters.

Specifically, all listed and detected clusters were dense as shown in Table 2. The three largest clusters had high densities. By inspecting reviewers in the three clusters, we found that the workers, who posted deceptive reviews on products advertised on crowdsourcing websites, belonged to the first two clusters. This indicates this clustering method worked well with the extended dataset. The last cluster contained legitimate reviewers who joined *Amazon Vine* program [1]. Users, who had posted high quality reviews and had gotten high helpfulness from other Amazon users, were invited by Amazon for the Vine program. We analyzed why these reviewers in the third cluster had high density, and found that they posted reviews of similar sets of products in a short time window. Interestingly, these reviewers posted many favorable reviews. The biggest difference between the first two clusters (deceptive reviewers) and the third cluster (legitimate reviewers) is that the legitimate reviewers in the third cluster posted lengthier reviews so that they can share detailed opinions regarding products that they had used.

Finally, we list a number of products associated with deceptive reviews written by the deceptive reviewers in the first two clusters. Table 3 shows the top-15 products favorably reviewed by the deceptive reviewers. Popular target products were health and beauty products, and books.

In summary, we evaluated our clustering method over both a synthetic dataset and a the extended (real) dataset. Our experimental result showed that the clustering method successfully found clusters of deceptive reviewers.

4.2 Detecting Deceptive Reviews

Given these encouraging results, we now turn to the challenge of detecting deceptive reviews. We adopt a standard

Table 2: Statistics and characteristics of detected reviewer clusters.

Cluster	Size	Avg. Weighted Degree	Characteristics
C1	1,079	14.5	Posted deceptive reviews for the same set of mostly health and beauty products
C2	285	31.1	Posted deceptive reviews for a set of meditation books
C3	1,273	13.0	Users of Amazon Vine program who do not write deceptive reviews
Everything Else	5,217	2.3	

Table 3: Titles of top 15 products which were favorably reviewed by deceptive reviewers.

Product Title	Product Category
maXreduce - Guaranteed Weight Loss	Health and Beauty
Krill Oil	Health and Beauty
Best Eye Cream	Misc.
Teeth Whitening	Health and Beauty
Tao II: The Way of Healing, Rejuvenation, Longevity, and Immortality	Hardcover
Omega 3 Fish Oil	Health and Beauty
Meditation: How to Reduce Stress, Get Healthy, and Find Your Happiness...	Paperback
Vitamin D3	Health and Beauty
Tao Song and Tao Dance: Sacred Sound, Movement, and Power...	Hardcover
Tao Song and Tao Dance (Soul Power)	Kindle Edition
Memory Loss	Health and Beauty
Green Tea	Health and Beauty
Soul Wisdom: Practical Treasures to Transform Your Life	Hardcover
Tao II: The Way of Healing, Rejuvenation, Longevity, and Immortality	Kindle Edition
Tao II: The Way of Healing, Rejuvenation, Longevity, and Immortality	Audible Audio Edition

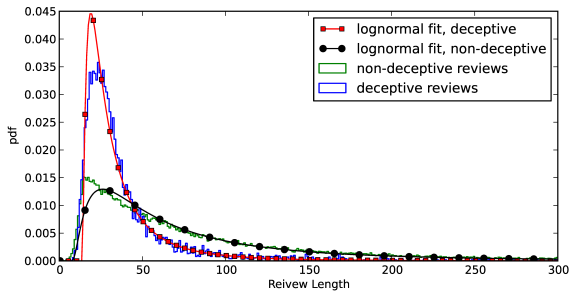

Figure 8: Length distributions of deceptive reviews and legitimate reviews fit log-normal distributions.

Support Vector Machine (SVM) with RBF kernel. We compare a deceptive review classifier using a standard set of features versus one that additionally incorporates the output of the proposed clustering method. We then compare the cluster-aware deceptive review classifier versus one based on an alternate clustering method (SimRank + k-medoids).

Ground Truth. To get the ground-truth (i.e., which review is deceptive or not), we labeled a review as deceptive if (i) the review was associated with a product which was targeted by a crowdsourced malicious task (that is, it was identified in the original *root dataset*); and (ii) the review with a high rating was posted within a few days after the task was posted. Otherwise, it is labeled as a legitimate review. Note that this labeling choice is conservative, in that there may be deceptive reviews that are labeled as legitimate. If a user posted at least one deceptive review, we considered the user as a deceptive reviewer.

Table 4 shows our collected dataset which contains 5.5K deceptive reviews and 65.6K legitimate reviews with a number of corresponding reviewers and products.

Table 4: Dataset.

	|Reviews|	|Reviewers|	|Products|
Deceptive	5.5 K	1.5 K	1 K
Legitimate	65.6 K	13 K	47.9 K

Features. To train and test a classifier, first we converted each review to a set of feature values. In this study, we used seven standard features plus one additional feature based on the output of our reviewer clustering method (see Table 5). The seven standard features are: *1. Verified purchase* feature is a binary feature. If a reviewer actually purchased a product that he reviewed, the feature value would be true. *2. Star rating* is the number of stars given to the product in the review. *3. Review length* is the logarithm of the length of the review in characters. We hypothesize that deceptive reviewers (crowd workers) would not want to spend a long time to write a review. In order to verify this, we compared length of deceptive reviews with length of legitimate reviews. Figure 8 shows length distributions of deceptive reviews and legitimate reviews, which follow log-normal distributions.

The log-normal distributions were also observed in other user generated text [24]. These distributions suggest that length of deceptive reviews are shorter than length of legitimate reviews. *4. Helpfulness ratio* is the number of helpful votes divided by all votes. On Amazon.com, a review has a star rating (1 through 5), and other users can rate the review as either "helpful" or "not helpful". We measured *helpfulness ratio* of each review in our dataset as follows: $\frac{\text{helpful votes}}{\text{helpful + not helpful}}$. Then, we grouped reviews with the same star rating for deceptive reviewers and legitimate reviewers. Figure 9 depicts helpfulness ratios with error bars of deceptive and legitimate reviews under star ratings. A circle and rectangle indicate a median helpfulness ratio of reviews. In star ratings $1 \sim 3$, deceptive and legitimate reviewers had

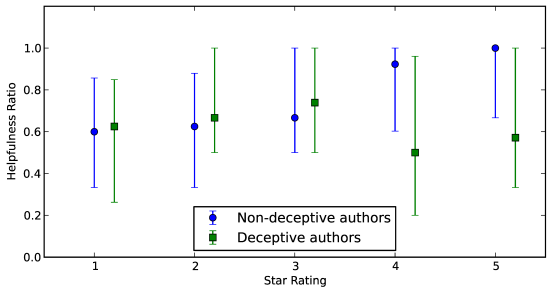

Figure 9: Helpfulness ratio of deceptive reviews and legitimate reviews under different star ratings.

similar helpfulness ratios. But, in $4 \sim 5$, deceptive reviewers' reviews got lower helpfulness ratios compared with ones of legitimate reviewers. This analysis reveals that even though legitimate users were able to punish some of these deceptive reviews with lower helpfulness ratings, many deceptive reviews were not so identified. *5. Total votes* is the denominator of *helpfulness ratio*, and captures the number of votes a review has accumulated. *6. Reviewer got more helpfulness for favorable reviews* is a binary feature that is true when favorable reviews (i.e., star rating is 4 or 5) of a reviewer are more helpful than his unfavorable (i.e., $1 \sim 3$)reviews. *7. Reviewer has more verified purchases for favorable reviews* is another binary feature.

We evaluate the quality of a deceptive review classifier based on these seven features alone versus a classifier based on these seven features plus the output of the proposed reviewer clustering method (from Section 3.2). This eighth feature is: *8. Cluster assignment:* the $K+1$-dimensional feature vector of indicator variables for whether the review author belongs to each of the clusters. The extra cluster is for when the reviewer is missing from the Reviewer-Reviewer Graph. By considering this feature separately, we can evaluate the importance of uncovering "hidden" connections among reviewers.

Unbalanced and Balanced Training and Testing Sets: From our dataset presented in Table 4, we created two pairs of training and testing sets: (i) unbalanced training and testing sets; and (ii) balanced training and testing sets. In the unbalanced training and testing sets, each set contains the half of the original dataset, following the same class ratio. But, to create balanced training and testing sets, first we performed undersampling of legitimate reviews so that the balanced dataset contains the same number of deceptive and legitimate reviews (i.e., 5.5K deceptive and 5.5K legitimate reviews). Then, we split the balanced dataset to the balanced training and testing sets each of which contains the half of the balanced dataset.

Evaluation Metrics and Setup: To evaluate prediction accuracy of our approach, we use three metrics: Area under the ROC Curve (AUC), precision and recall. Especially, we use AUC as our primary evaluation metric, since a higher AUC means that a model is good at correctly predicting both class instances regardless of class imbalance [8].

We developed two SVM classifiers: (i) a basic SVM classifier based on the first seven features in Table 5; and (ii) an advanced SVM classifier based on all eight features including the output of our clustering method. The basic classifier was used as a baseline. Each classifier (e.g., basic and advanced) was trained by each of training sets.

Table 5: Features of our deceptive review classifier.

1. Verified purchase
2. Star rating
3. Review length
4. Helpfulness ratio
5. |helpful + unhelpful votes|
6. Reviewer got more helpfulness for favorable reviews
7. Reviewer has more verified purchases for favorable reviews
8. **Cluster assignment**

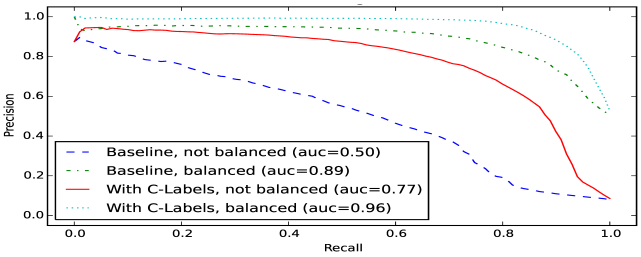

Figure 10: Precision and recall curves of review classification results under four cases (2 different datasets and 2 different classification approaches).

Results: Figure 10 shows experimental results of two classifiers with unbalanced and balanced training and testing sets. In case of using unbalanced training and testing set, while the basic SVM classifier achieved 0.50 AUC, the advanced SVM classifier achieved 0.77 AUC. In case of using balanced training and testing set, AUC has been increased in both basic and advanced SVM classifiers. Specifically, while the basic SVM classifier achieved 0.89 AUC, the advanced SVM classifier achieved 0.96 AUC. Overall, adding the output of our clustering method to feature set significantly improved AUC by 54% and 8% in both unbalanced and balanced data sets. Such performance improvement indicates the effectiveness of incorporating the social context of reviewers (i.e., the output of the clustering method) into deciding whether a review is deceptive or legitimate.

Versus an Alternative Clustering Method: Finally, we investigate the quality of deceptive reviewer classification if the proposed clustering method is swapped out in favor of an alternate one. Concretely, we consider a clustering method based on SimRank [12] and K-medoids. First, SimRank measures similarity between nodes of a graph. Then, k-medoids clustering algorithm is performed based on the similarities of nodes in the graph. Specifically, we calculated SimRank score in two graphs – (i) a bipartite graph containing both reviewers and products; and (ii) the Reviewer-Reviewer collaboration graph $G_{R,R}$. After a similarity matrix of each graph is calculated, we use k-medoids to cluster the reviewers for each graph. Finally, a cluster membership of each reviewer is used as a feature for a SVM based deceptive review classifier. In this experiment, we used the unbalanced training and testing set.

Since we had two different similarity matrices, we built two classifiers. The first clustering method based on the bipartite graph performed poorly enough to be almost identical to the performance of the seven features based SVM classifier without using any clustering output. The second clustering algorithm based on the Reviewer-Reviewer graph achieved 0.71 AUC which was lower than performance (0.77 AUC) of our proposed approach as shown in Figure 11.

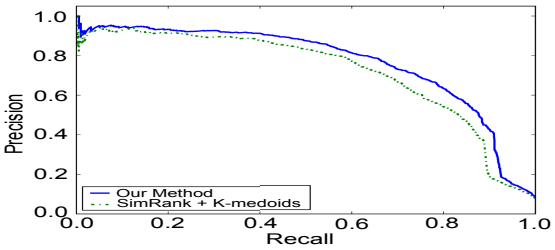

Figure 11: Our classification approach with the cluster assignment (0.77 AUC) outperformed SimRank and k-medoids based classifier (0.71 AUC) under unbalanced dataset.

With respect to efficiency, SimRank took a long time O(hours) for a large graph since time complexity of the original SimRank is $O(n^4)$ where n is the number of the nodes in a graph compared to our pairwise MRF parameterization which has $O(n^2)$ parameters and took O(minutes).

In summary, our proposed clustering based classifier outperformed SimRank and k-medoids based classifier in terms of both effectiveness and efficiency.

5. CONCLUSION

In this paper, we presented a novel sampling method for identifying products that have been targeted for manipulation and a seed set of deceptive reviewers who have been enlisted through the command-and-control of crowdsourcing platforms. Specifically, we have sampled tasks targeting Amazon via the crowd marketplaces RapidWorkers.com, ShortTask.com, and Microworkers.com. We have augmented this seed set to identify additional deceptive reviewers who participate as part of hidden groups of coordinated manipulation through a reviewer-reviewer graph clustering approach. Finally, we have embedded the results of this probabilistic model into a classification framework for detecting crowd-manipulated reviews. Our classification approach using the reviewer clustering results as a feature significantly outperformed a classification approach not using the reviewer clustering results. Specifically, our approach with clustering results have achieved 0.77 AUC in unbalanced testing set and 0.96 AUC in balanced testing set, improving 54% AUC in unbalanced testing set and 8% AUC in balanced testing set respectively compared with the classification approach without using reviewer clustering results. Additionally, the proposed approach has outperformed a SimRank and K-medoids based approach in terms of effectiveness and efficiency.

6. ACKNOWLEDGEMENTS

This work was supported in part by AFOSR Grant FA9550-12-1-0363 and Google Faculty Research Award. Portions of Dr. Squicciarini's work was supported by the Army Research Office under grant W911NF-13-1-0271. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

7. REFERENCES

[1] Amazon. Amazon vine program. `http://www.amazon.com/gp/vine/help`, August 2014.

[2] D. Anguelov, D. Koller, H.-C. Pang, P. Srinivasan, and S. Thrun. Recovering articulated object models from 3d range data. In *UAI*, 2004.

[3] M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich. Soylent: A word processor with a crowd inside. In *UIST*, 2010.

[4] J. A. Chevalier and D. Mayzlin. The effect of word of mouth on sales: Online book reviews. Technical report, National Bureau of Economic Research, 2003, `http://www.nber.org/papers/w10148`.

[5] C. Danescu-Niculescu-Mizil, G. Kossinets, J. Kleinberg, and L. Lee. How opinions are received by online communities: A case study on Amazon. com helpfulness votes. In *WWW*, 2009.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39:1–38, 1977.

[7] C. Elliott. Hotel reviews online: In bed with hope, half-truths and hype. `http://www.nytimes.com/2006/02/07/business/07guides.html`, Feb, 2006.

[8] T. Fawcett. An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861–874, June 2006.

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

[10] C. Forman, A. Ghose, and B. Wiesenfeld. Examining the relationship between reviews and sales: The role of reviewer identity disclosure in electronic markets. *Information Systems Research*, 19(3):291–313, 2008.

[11] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: Answering queries with crowdsourcing. In *SIGMOD*, 2011.

[12] G. Jeh and J. Widom. Simrank: A measure of structural-context similarity. In *KDD*, 2002.

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

[14] J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *Journal of the ACM (JACM)*, 49(5):616–639, 2002.

[15] K. Lee, J. Caverlee, Z. Cheng, and D. Z. Sui. Campaign extraction from social media. *ACM Trans. Intell. Syst. Technol.*, 5(1):9:1–9:28, Jan. 2014.

[16] K. Lee, B. D. Eoff, and J. Caverlee. Seven months with the devils: A long-term study of content polluters on twitter. In *ICWSM*, 2011.

[17] K. Lee, P. Tamilarasan, and J. Caverlee. Crowdturfers, campaigns, and social media: Tracking and revealing crowdsourced manipulation of social media. In *ICWSM*, 2013.

[18] S. Lee and J. Kim. Warningbird: Detecting suspicious urls in twitter stream. In *NDSS*, 2012.

[19] Y. Lu, P. Tsaparas, A. Ntoulas, and L. Polanyi. Exploiting social context for review quality prediction. In *WWW*, 2010.

[20] C. C. Miller. Company settles case of reviews it faked. `http://www.nytimes.com/2009/07/15/technology/internet/15lift.html`, Jul, 2009.

[21] A. Mukherjee, B. Liu, and N. Glance. Spotting fake reviewer groups in consumer reviews. In *WWW*, 2012.

[22] M. Ott, C. Cardie, and J. Hancock. Estimating the prevalence of deception in online review communities. In *WWW*, 2012.

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

[24] P. Sobkowicz, M. Thelwall, K. Buckley, G. Paltoglou, and A. Sobkowicz. Lognormal distributions of user post lengths in internet discussions-a consequence of the weber-fechner law? *EPJ Data Science*, 2(1):1–20, 2013.

[25] D. Streitfeld. In a race to out-rave, 5-star web reviews go for $5. `http://www.nytimes.com/2011/08/20/technology/finding-fake-reviews-online.html`, Aug, 2011.

[26] VIP deals rebate letter. `https://www.documentcloud.org/documents/286364-vip-deals.html`, Dec, 2011.

[27] G. Wang, T. Wang, H. Zheng, and B. Y. Zhao. Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers. In *USENIX Security*, 2014.

[28] G. Wang, C. Wilson, X. Zhao, Y. Zhu, M. Mohanlal, H. Zheng, and B. Y. Zhao. Serf and turf: Crowdturfing for fun and profit. In *WWW*, 2012.

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