DETECTING CROWDSOURCED SPAM REVIEWS IN SOCIAL MEDIA

A Thesis

by

AMIR ALI FAYAZI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	James Caverlee
Committee Members,	Ricardo Gutierrez-Osuna
	Rishika Rishika
Head of Department,	Duncan M. Walker

December 2013

Major Subject: Computer Science

Copyright 2013 Amir Ali Fayazi

ABSTRACT

User submitted reviews are used by potential buyers to evaluate products before their purchase. In this work we study cases of deceptive reviews on Amazon.com which rate the products favorably. These were paid for through a number of crowdsourcing websites. The behavior of the review spammers as a group has distinguishable characteristics which are used in our proposed method. We use a probabilistic model for spammer pairwise collaboration which is used to cluster reviewers. The introduced model is verified on a set of synthetic data and outperforms a baseline classifier which treats reviews on their own, without their social context. The performance of the proposed method for detecting clusters of spammers is also compared to an alternative approach. Finally we demonstrate some of the detected clusters of review spammers on the data set which was crawled from Amazon.

ACKNOWLEDGEMENTS

First, I would like to thank my parents and family for providing for my education throughout the years. I also would like to express my gratitude to my adviser, Dr. Caverlee, for giving me directions, feedbacks, and insights during my research.

Finally, I would like to thank members of the Infolab: Elham, Yuan, Zhiyuan, Kyumin, and all other members from whom I learned valuable lessons and advice.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Online reviews are ubiquitous and assist buyers in making purchase decisions. Websites such as Yelp.com and Epinions.com have been running a viable business around them. The effect of online product reviews on its sales has been demonstrated in the literature [3, 8].

As a result, there is incentive for polluting the reviews with spam/deceptive reviews to get a product promoted. The deception in online reviews is a growing problem [20, 26, 29]. For instance, up to 6% of the reviews on websites such as Yelp or TripAdvisor are estimated to be deceptive/spam [20]. Even more troubling is the rise of *crowd-sourced* spam. Crowd-sourcing is the delegation of a task to a large group of workers usually over the Internet as opposed to traditional employees or suppliers. In [29] Wang et al. demonstrate evidences of a business behind some of the crowdsourced astro-turfing where a merchant who wants to have its product promoted interacts with an agent whose job is to pay a set of Internet workers posting spam content to promote the product. They estimate $4 million having been spent only in the crowdsourcing websites they studied.

Polluting user-generated reviews by deceptive ones results in online reviews losing the consumers trust or worse, it could inflict harm on the consumers if the product does not meet standard safety requirements, especially medical products. For instance [6] reports a business traveler who became skeptical of the reviews she sees online. "I read reviews of hotels that I've stayed at," she said. "And they're just wrong. I wonder if they've really stayed at the hotel." The reporter interviewed few consumer review website owners. HotelShark and IgoUgo, two small review websites simply vet every posted reviews. Larger websites such as TripAdvisor hired

implement fraud detection algorithms and punish hotels which try to manipulate the ranking. Major hotel chains like InterContinental and Hilton also pay attention to what it is said about them on the Internet. In another case, [26] reports on how a Kindle accessory merchant asked customers who bought their product to leave a five star review in return for a rebate [28]. This resulted in a stream of 5-star reviews on the product page. Amazon later deleted those reviews and the product was removed from the store subsequently. There are numerous other similar cases of deceptive reviews written to promote a product or service [18, 27]. Among the reviews we collected there are cases where the consumers was subject to physical harm the medicine whose rating was promoted through deceptive means. For instance, for a weight-loss pill, a customer wrote "... 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...".

Hence, in this thesis we focus on the emerging problem of detecting crowdsourced spam reviews in social media. A method to filter out spam and deceptive reviews provides for higher overall quality of the reviews and also benefits other application such as sentiment analysis and extraction and review summarization. We have observed cases where crowdsourcing websites are used to pay workers to write favorable reviews for products on Amazon.com. A sample of such tasks is shown in Figure 1.1 and Figure 1.2 shows a sample of deceptive reviews written in response to such tasks. Such crowdsourcing websites often practice minimal moderation on the types of tasks being submitted.

A common challenge in review spam literature is obtaining ground truth for deceptive or spam reviews. Access to a set of products which explicitly asked for deceptive reviews allowed us to build a labeled ground truth for a set of deceptive reviews. In addition, since crowdsourced spam is generated by humans rather than

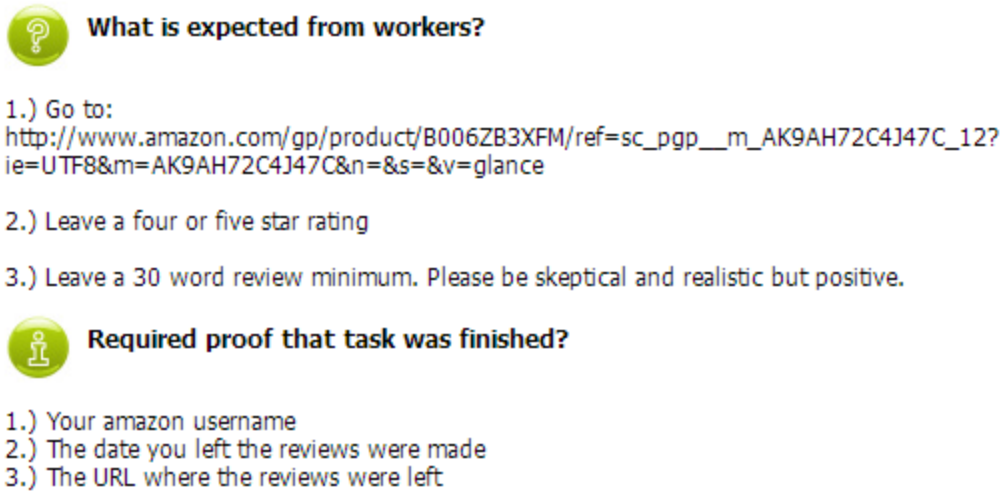

Figure 1.1: Sample crowdsourced task which pays off $0.25

computers, their detection is more challenging as humans can actively circumvent the detection measures. For instance, a simple signal to review quality on Amazon.com is the helpful votes it has received. However, there are crowdsourced tasks to up-vote spam reviews.

Having known a set of spam and non-spam reviews allowed us to analyze the characteristics of both the reviews and the authors of those reviews. We noticed those who write deceptive reviews generally do so in groups. This does not necessarily mean the authors of spam reviews are aware of one another but is more likely the result of the process of soliciting favorable reviews. A tasks which generally pays a small sum of money per each favorable review is submitted on a crowdsourcing website. Once submitted, a group of workers who frequent that website end up fulfilling it within a relatively short time period. Based on this, we focused on identifying review spammers as it allows us to utilize the social context of deceptive reviews. Reviews of products are not independent entities hence use of the social context in evaluating their helpfulness has already proved helpful in the literature [16, 19].

Advanta Supplements Omega3 Fish Oil, 60 Softgels (Pharmaceutical Grade Omega-3)
Offered by Advanta Supplement
Price: $19.67

1 of 3 people found the following review helpful

★★★★☆ **Omega 3 fish oil**, September 17, 2011

This review is from: Advanta Supplements Omega3 Fish Oil, 60 Softgels (Pharmaceutical Grade Omega-3) (Health and Beauty)

great product, works great for lowering cholesterol, relief from arthritis and joint pain, leaves your hair shiny and leaves no after taste in your mouth.

Comment | Permalink

Working For You: Developing Your Career by Becoming the Best You Can
by Mark Clayson
Edition: Paperback
Price: $8.54

★★★★★ **developing your career**, September 17, 2011

This review is from: Working For You: Developing Your Career by Becoming the Best You Can (Paperback)

Very informative and educative read for anyone trying to shape thier careers and striving to become the best they could. well thought out and sourced, brilliant read

Comment | Permalink

How to Save $100's of Dollars on Your Grocery Bill
Price: $2.99

★★★★★ **cut your groceries bill in half using coupons**, September 16, 2011

This review is from: How to Save $100's of Dollars on Your Grocery Bill (Kindle Edition)

I guess you never how valuable a book can be until you've read or even better how amazingly you could be saving on your grocery bills with money saving coupons, I have saved over $300 what a credit cruncher wow!

Comment | Permalink

Figure 1.2: A set of spam/deceptive reviews written by a user

1.1 Outline of This Thesis

Initially we form a dataset of reviews posted in Amazon and label the reviews written in response to a crowdsourced task as a spam (deceptive) review. A graph of reviewers is then built where nodes are people (review authors) and connections are *collaboration* in writing deceptive spam reviews. Our hypothesis is that, in this graph, those reviewers who get paid by the crowdsourcing websites have a higher chance of writing a review for a product in the same time frame. Thus, in the reviewer collaboration graph they tend to have more connections than a set of random reviewers. As a result, they should form clusters in that graph. To do the clustering, we make use of a Markov Random Field based on this author collaboration graph. It takes into account individual reviewer characteristics and pairwise collaborations. The parameters for the MRF is learned using the Expectation Maximization (EM) framework. The result of learning is a partitioning of users into clusters. Now if a reviewer is known to write deceptive reviews, it can be generalized to all the reviewers in its cluster. During this task, certain features of both the reviews and reviewers are used. They are selected based on our analysis of the labeled dataset of reviews.

The clustering method is tested on a synthetically generated graph of collaborating reviewers. Next we cluster reviewers of a real world dataset of Amazon reviews. Once reviewers are clustered, we make use of the labeled (deceptive vs. non-deceptive) reviews to distinguish and demonstrate how using the cluster assigment of the review authors as a feature helps the classification of reviews into spam/non-spam significantly. A list of spammer clusters found in our dataset is presented afterwards. Finally, we compare our method to an alternative competing method which employs a different approach to clustering review spammers.

In the following section a set of relevant works are discussed. Next, in Section 3,

the dataset we used is described along with the analysis on the data. Section4 contains the problem formulation and the proposed model. Finally, in Section 5, the results of the model evaluation is presented.

2. RELATED WORKS

Wang et al. [29] analyzed the use of crowdsourcing websites for malicious activities. They also submitted crowdsourced astroturfing (*crowdturfing*) tasks of their own. Their posted tasks received responses by the crowdsourcing websites workers within few hours to few days. They conclude that crowdturfing tasks are widespread and bring about considerable revenue. Their study also revealed double digit growth of spam campaigns on the websites they studied. Ott el al. [20] also found deceptive reviews to be growing in size in a number of consumer review websites they studied.

In order to detect deceptive reviews Mukherjee et al. [19] found it easier to classify groups of spammers rather than individual spammers. Among the features they used, those of groups of spammers had the most distinguishing power while linguistic features from the review text had the least. They used human judges to label reviewers who post deceptive reviews. Findings of Ott et al. [21] also agree with the fact that linguistic features are not reliable per se as it is even difficult for humans to discern between a deceptive and non-deceptive review.

Gao et al. [9] analyzed Facebook wall posts with URL to detect coordinated spam campaigns. They built a similarity measure on the URLs first and built a graph of URLs connected if they are similar. Finally the connected subgraphs are marked as spam if they are *bursty* and *distributed* that is if the URLs were posted in a short time and from different users.

Usage of social context of reviews is not only useful to finding groups of spammers, Lu et al. [16] show one can assess the quality of consumer reviews better (especially for small training data size) by making the assumption that the quality of a reviewer depends on the quality of its peers in the social network of review-

ers. Danescu-Niculescu-Mizil et al [4] suggest a simple underlying model for review quality (helpfulness). They show when controversy on a consumer product is high, reviews whose rating diverge from the average rating of the product often get higher helpfulness scores.

Even though detecting deceptive/spam reviews on their own is a challenging task even for humans, there has been attempts to discover patterns in the rating [7] or temporal [30] distribution of spam reviews. Li et al [15] classified reviews into spam and non-spam with limited success. They made use of various features including sentiment scores, product popularity and reviewers ranking.

3. DATA SET

In this section we describe our data set and how it was obtained. Currently, there is no standard data set of crowdsourced spam reviews to the best of our knowledge. A number of crowdsourcing websites do not practice much moderation [13, 17, 22, 24]. Over the course of several weeks, we crawled [22, 24, 17] for tasks which requested four or five star reviews for a product on Amazon.com. We gathered all such products and people who had written favorable reviews for the product within few days of the time the task was posted. We associated such products and review authors with deceptive reviews. See Figure 3.1 and Figure 3.2.

We extended our data set to incorporate normal (non-deceptive) reviews into it. We continued crawling Amazon in a breadth first fashion from the products and people who engaged in writing deceptive reviews. That is, for each product we crawled all people who reviewed it and for each person, we crawled all the products reviewed by that person. We continued the crawling in BFS for three levels. By crawling this way, for each person and product in our dataset (except for the fringe of BFS) the complete set of reviews associated with them was obtained. See Table 3.1.

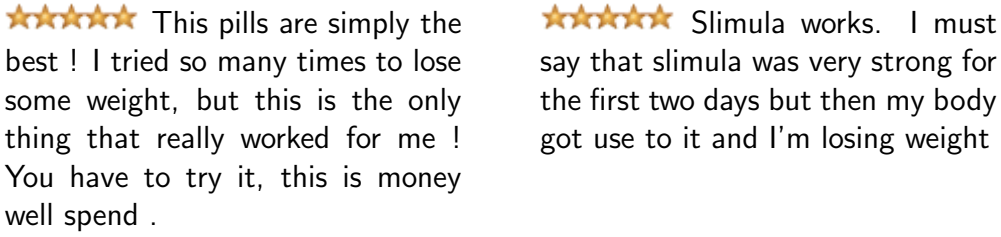

Figure 3.1: Two cases of deceptive reviews for a product

★☆☆☆☆ . . . I suspect these reviews are fake for multiple reasons. First, they all read like they were written by the same person. Secondly, it looks like they were all posted right on top of each other. Third, if you look into the previous reviews written by these reviewers, you will see a TON of the same products! It is a bit suspect when you have 40 reviews appear in the first two weeks of selling this product, and no reviews for a month after that?!? Quickly followed by another 20 reviews in a second one week time span? Not very likely.

Figure 3.2: Reviewer doubts the authenticity of other deceptive reviews

Table 3.1: Dataset statistics

	Reviews	**People**	**Products**
Crawl Result	118.1 K	19.7 K	75.5 K
Cleaned Data	71 K	14.5 K	46.9 K
Deceptive	5.5 K	1.5 K	1 K
Non-Deceptive	65.6 K	13 K	47.9 K

3.1 Analysis on the Dataset

The dataset was analyzed during which we did some feature engineering. Few features are suggested and we show how they can individually distinguish between deceptive and non-deceptive reviews. The reviews are labeled as deceptive, only if both the product and the author of the review are labeled as deceptive.

On Amazon.com, a review has a star rating (1 through 5) and optionally the number of people who found the review helpful vs. not helpful. We call $\frac{\text{helpful votes}}{\text{helpful + not helpful}}$ the *helpfulness ratio*. In Figure 3.3 for each star rating, the median helpfulness ratio of reviews having that rating is displayed along with error bars indicating second and third quantiles. The spammers and non-spammers have similar helpfulness ratios for unfavorable or neutral (1 through 3 stars) reviews.

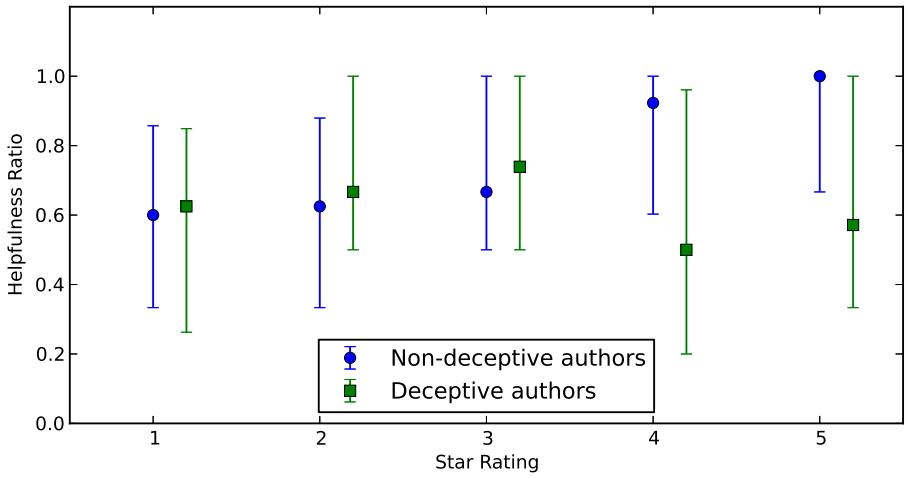

Figure 3.3: Star rating vs. helpfulness ratio

However, spammers received considerably lower helpfulness ratios for their favorable reviews compared to non-spammers. This suggest, firstly, that the deceptive reviews are not generally negative advertising. This is the result of the fact that tasks posted on crowdsourcing websites we used to label our dataset asked for a favorable reviews for their products. Secondly, the favorable (4, 5 star) reviews from non-spammers have a denser distribution of helpful ratios compared to that of spammers. This could be due to the fact that there are tasks on crowdsourcing websites which ask for up-voting the helpfulness of deceptive reviews. This creates some cases of deceptive reviews with high helpfulness votes yet does not change the overall skew.

Another piece of information obtained throughout our crawl is whether the person writing the review has purchased the product through Amazon.com. Intuitively, it does not make sense for review spammers to actually purchase the product over Amazon as it incurs a high cost per review. Still, we observed very few cases where the merchant purchased its own products and wrote favorable reviews.

In Figure 3.4 it can be seen that for those who write deceptive reviews, a favorable review corresponds to low chances of actually having purchased the product while the case for others who don't write deceptive reviews is the opposite; The probability of purchase for non-spammers increases slightly for favorable reviews.

The last distinguishing feature we observed was the length of the reviews in characters. The hypothesis is that writing elaborate and lengthy reviews takes too much effort to be worth it given the low pay off each deceptive review earns from the crowdsourcing websites. Hence deceptive reviews should be shorter.

In our dataset, the review lengths approximately followed log-normal distribution (Figure 3.5). This distribution has also been observed in other user generated text [25]. For each review author, we consider the average log-length of his reviews which with an independence assumption also follows Gaussian distribution. In Figure 3.6 the distribution of review length of deceptive authors is more concentrated around a lower mean value while that of non-deceptive authors are more spread out but has a higher mean. This conforms to our hypothesis about review lengths being different in two cases.

We avoided using features relying on the review textual content as such features have not been shown to be useful in the previous studies and can add more noise than signal. It is non trivial for even human judges to discern between deceptive and non-deceptive reviews based solely on their content [12].

3.2 Labeling the Data

A review is labeled deceptive if both of these conditions are met: *a*) The product being reviewed solicited reviews on the crowdsourcing websites. *b*) The review author wrote a positive review for the product within a short time from when the crowdsourcing task was posted. Otherwise it is labeled as non-deceptive.

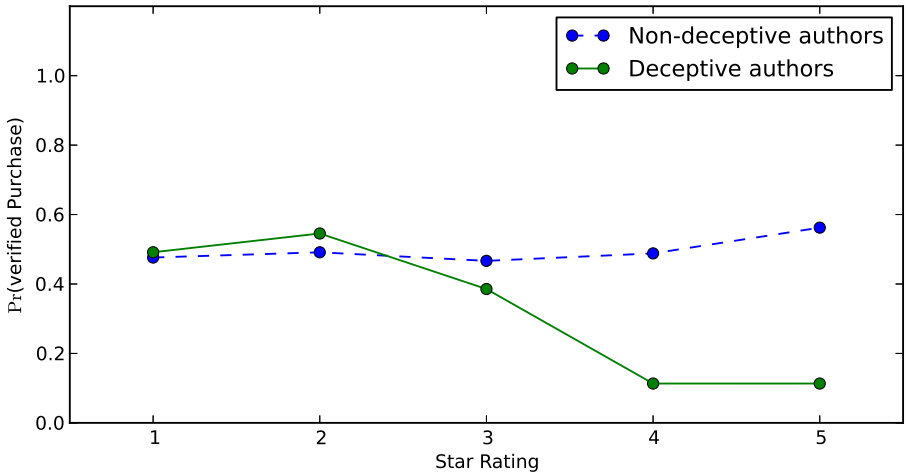

Figure 3.4: Star rating vs. Pr(Verified Purchase)

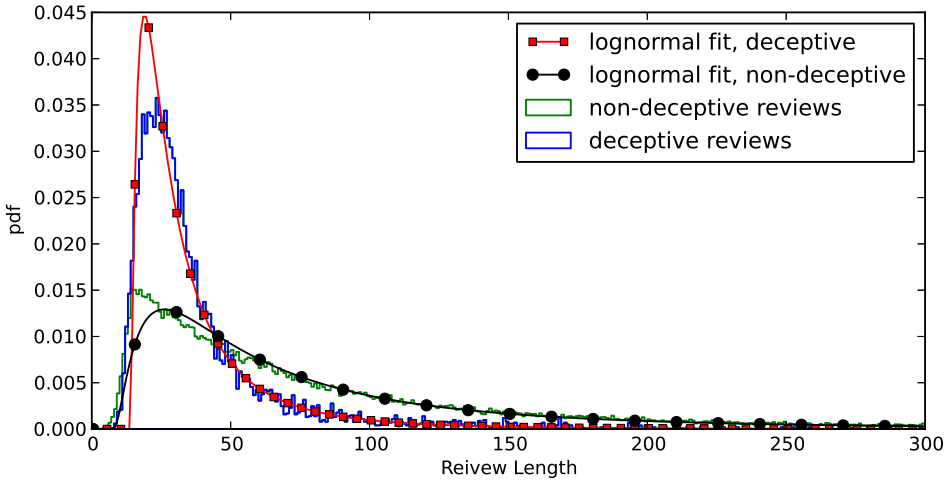

Figure 3.5: Fitting lognormal to length distribution

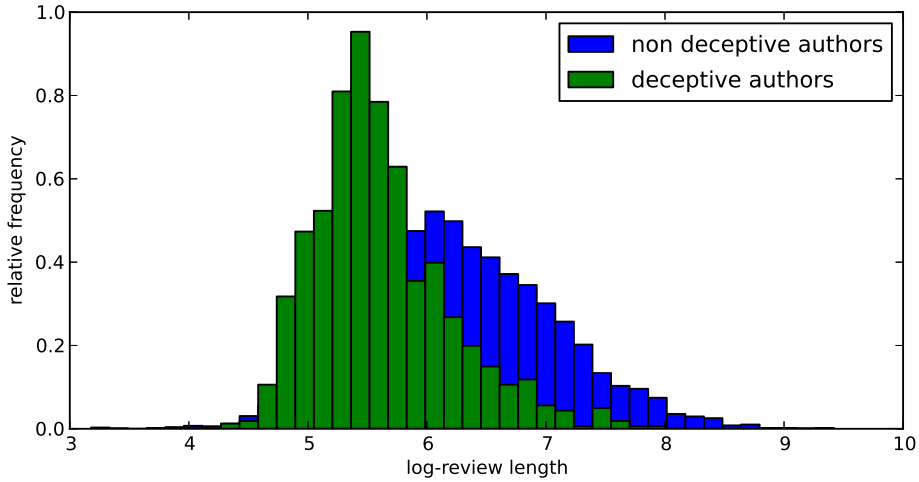

Figure 3.6: Review lengths distribution of deceptive authors vs. non-deceptive authors

4. PROPOSED METHOD

In this section we propose a method to detect spam reviews. To do so, we cluster the review authors first. Then use the cluster assignment of the review author among other features in a review classifier. In the next section we demonstrate how this single cluster label feature affects the classification performance significantly. The clustering method is also devised to be tailored to this problem which is described later in this section.

4.1 Reviews as Standalone Documents

We initially built a baseline classifier which takes reviews as standalone data points. For this purpose, each review is represented by the features listed in Table 4.1. The meanings of the features in Table 4.1 is as follows. The feature *Verified Purchase* is binary and is true if an actual purchase was made through Amazon for the product being reviewed. *Star Rating* is the number of stars given to the product in the review. *Review Length* is the logarithm of the length of the review in characters. *Helpfulness ratio* is the number of helpful votes divided by all votes and the next feature is the denominator of the *helpfulness ratio*. The binary variable *more helpfulness for favorable reviews* is true when favorable reviews of the author of the review are more helpful than his unfavorable reviews and finally the binary variable *Author has more verified purchases for favorable reviews* is self-explanatory.

We form a supervised learning using the labels described in Section 3.2. For the classification we used Support Vector Machine (SVM) with Radial Basis Function (RBF) kernels with half the dataset used for training and half for test. The number of deceptive reviews is about 5.5K and non-deceptive reviews is about 65.6K (see Table 3.1). There is an imbalance in the class sizes. Thus the classification was run

Table 4.1: Features of reviews used in classification

Verified Purchase
Star Rating Review Length
Helpfulness Ratio
Total # helpful + unhelpful votes
Author is more helpful for favorable reviews
Author has more verified purchases for favorable reviews

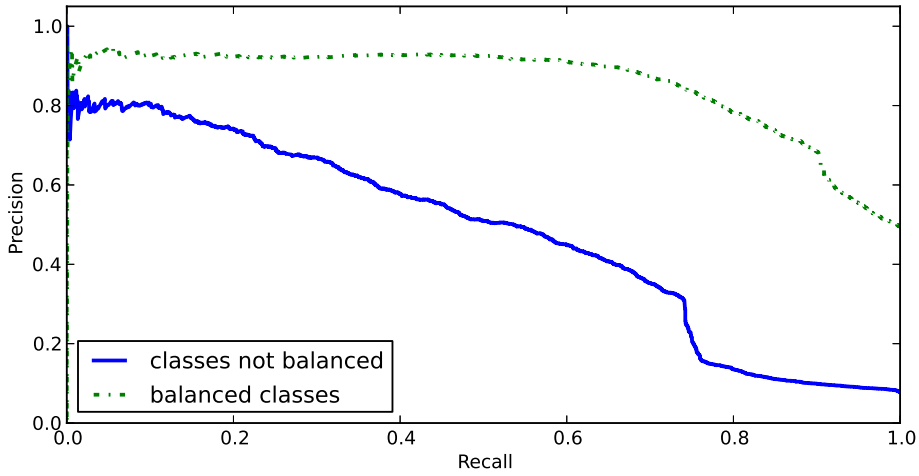

Figure 4.1: Standalone review classification performance

both on the original set and a balanced dataset obtained by discarding members of the larger class until it matches the size of the smaller one.

The performance for the balanced classes are significantly better (Figure 4.1). In the unbalanced dataset the classifier tends to favor non-deceptive classification which sacrifices recall for accuracy. Considering that deceptive reviews comprise a small portions of all reviews on Amazon, real world performance of a naïve classifier based on review features might not be very effective.

4.2 Reviews in Their Social Context

By using the social context of reviews we try to find users who write deceptive reviews in coordination and use that information to label reviews as deceptive. This is in contrast to treating each review independently. The payoff for a single favorable review generally less than a US dollar. Thus, it also makes sense for each user to write more than one such reviews.

The dataset can naturally be represented as an Author-Product graph $G_{A,P}$. It is a bipartite graph of consumer products on one part and authors on the other with edges representing reviews. We form an author-author graph $G_{A,A} := (A, E_{A,A})$ where the vertex set A are authors who have written reviews on Amazon.com and the weighted edges $E_{A,A}$ are how much collaboration each pair of reviewers have had during a short time window. This graph is obtained by projecting the Author-Product graph $G_{A,P}$ onto its *Author* nodes. That is, placing edge between authors if there is a common reviewed product. The edges weights are defined as follows:

$$w_{A,A}(a,b) := \frac{|\{p|p \in N_{A,P}(a) \wedge p \in N_{A,P}(b) \wedge |t(a,p) - t(b,p)| < W\}|}{\min\left(|N_{A,P}(a)|, |N_{A,P}(b)|\right)} \tag{4.1}$$

Where $N_{A,P}(a)$ is the set of products rated favorably (neighbors) by a. $t(a,p)$ is the time a reviewed p and finally W is a constant window size which was set to three days in our case. Basically, the numerator is counting the number of products both a and b rated favorably in the same time window.

4.3 Probabilistic Modeling

So far, the problem is to cluster authors in the Author-Author graph $G_{A,A}$ so those who have collaborated on writing deceptive reviews end up in the same cluster. In

this section we describe the clustering of Author-Author graph. The clustering has two distinct characteristics. First, if a pair of authors have collaborated heavily they tend to be assigned to the same cluster. This allows for a sort of 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 need a predetermined number of clusters either. On the other hand, this would merge two dissimilar clusters into one simply based on one edge between them. To counter this, all the authors of a cluster should have similar features to each other. For instance, if one cluster of authors all have actually purchased the product they have reviewed and another set of authors all have not purchased the products they have reviewed, they tend to be in separate clusters.

We formulate this problem using a Markov Random Field defined over $G_{A,A}$. Each node in the random field corresponds to a review author and is a discrete random variable whose value is the cluster the corresponding author belongs to. We aim to maximize Equation 4.2 which is the model likelihood. There, 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); and θ represent the parameters of the model. Since the MRF is wholly conditioned on the observation data D it is an instance of Conditional Random Field (CRF).

The likelihood is composed of two types of potential functions. The first type is per each author ϕ_j hence called singleton and the other is per pair of authors ϕ_{pair} hence called pair potential. The singleton potentials have higher value when the author features are close to that of the mean cluster features in the feature space. The pair potentials are higher when two connected authors on the graph are assigned the same cluster.

$$P(\boldsymbol{Z}|\boldsymbol{\theta}, \boldsymbol{D}) \propto \prod_j \phi_j(Z_j) \prod_{j,k} \phi_{\text{pair}}(Z_j, Z_k) \tag{4.2}$$

In Equation 4.2 the variable Z_j is the (hidden) cluster of author j. $\phi_j(Z_j)$ is the potential function over labeling of author j and ϕ_{pair} is over pairs of labeling. This modeling is similar to that of [1].

4.3.1 Singleton Potentials

The singleton potential function is defined as follows.

$$\phi_j(Z_j) = \Pr(Z_j, \overbrace{\boldsymbol{F_j}, \boldsymbol{P_j}}^{\text{Observations}} |\boldsymbol{\theta}) = \Pr(\boldsymbol{F_j}, \boldsymbol{P_j}|Z_j, \boldsymbol{\theta}) \Pr(Z_j|\boldsymbol{\theta}) \tag{4.3}$$

The variables $\boldsymbol{F_j}$ and $\boldsymbol{P_j}$ are the observed data and Z_j is the hidden variable (cluster of author j). $\boldsymbol{\theta}$ is the model parameters. $\boldsymbol{F_j}$s are the features of author j:

- Real Name: whether the name has been verified by Amazon through e.g. credit card - *Binary*
- Helpfulness of reviews – *Binary*
- Review is based on a verified purchase - *Binary*
- Length of the reviews written – *Log normal*
- Whether favorable of the author are more helpful – *Binary*
- Whether favorable of the author are based on a verified purchase – *Binary*

$\boldsymbol{P_j}$s are the products reviewed by author j.

We make the assumption that feature values are conditionally independent given the cluster. Figure 4.2 shows a graphical model of our independence assumption. Also, in Equation 4.4 the symbol $\sim$ stands for adjacency in the graph.

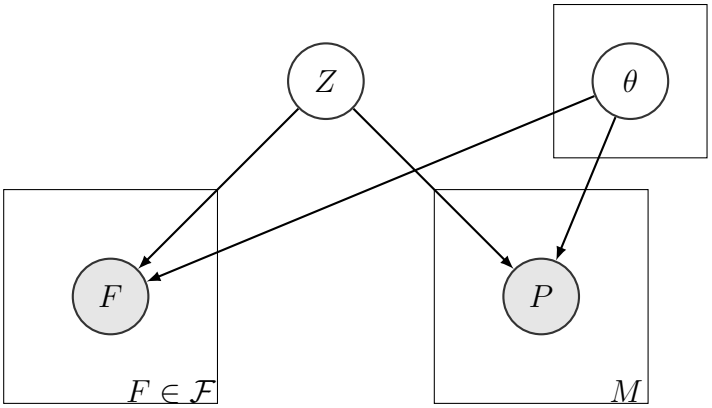

Figure 4.2: The independence assumption of the variables during calculation of singleton potentials ϕ. The indicator variables P correspond to each product and indicates whether that product was reviewed favorably. We assume the probability of that only depends on which cluster Z the author belongs to and the preference of that cluster θ. F represents various features of the authors in the cluster

$$\Pr(\boldsymbol{F_j}, \boldsymbol{P_j}|\theta, Z_j) = \prod_{F_{jk} \in F_j} \Pr(F_{jk}|Z_j, \theta) \prod_{j \sim k} \Pr(P_{jk}|Z_j, \theta) \tag{4.4}$$

The binary variables *Real Name*, *Helpfulness*, and *Verified Purchase* have Bernoulli distribution.

$$\Pr(F = f|Z = c) = \begin{cases} p_c & f = 0 \\ 1 - p_c & f = 0 \end{cases} \tag{4.5}$$

In Section 3.1 we argued that the length of reviews has a log-linear distribution. Also the average log-length of reviews of an author follows Gaussian distribution which is denoted by L in Equation 4.6.

$$\Pr(L = \log l|Z = c) = \mathcal{N}(\log l; \mu_c, \sigma) \tag{4.6}$$

We assume each cluster of coordinated authors do not review all products uni-

formly but have concentrated on writing reviews for a subset of them. 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 an author in cluster c. This part of the singleton potential tends to push dissimilar set of authors into separate clusters.

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

4.3.2 Pair Potentials

The other potential function of the Markov random field likelihood formulation is the one between pairs of authors. The role of this function is to force authors who have collaborated on writing favorable reviews end up in the same clusters. Also in case the number of clusters is larger than the true value, this part regulates the resulting clustering where too many clusters with intra-edges are punished. The likelihood of the similar cluster labels for two adjacent authors in the graph depends both on $0.5 < \tau < 1$ which is a tunable parameter of the model and on the weight of collaboration between the two authors $w_{A,A}(j,k)$.

$$\phi_{\text{pair}}(Z_j, Z_k) = \begin{cases} \tau^{w_{A,A}(j,k)} & Z_j = Z_k \\ (1-\tau)^{w_{A,A}(j,k)} & Z_j \neq Z_k \end{cases} \tag{4.7}$$

By increasing the value of τ, connected authors are more likely to be in the same cluster. The value of τ was fixed at 0.8 in our case.

4.4 Learning Parameters and Clusters

We have three sets of data:

- 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 $w_{A,A}$

We aim to maximize the likelihood $\mathcal{L}(\boldsymbol{Z}, \boldsymbol{\theta}; D)$. A common way to deal with maximizing likelihood functions which depend on hidden data (Z) is to use a local optimization methods like Expectation Maximization (EM). EM iterates over two steps throughout which the log-likelihood (hence the likelihood itself) is guaranteed to increase [5]. However it might get trapped in a local maximum. Therefore, we restart the algorithm several times from random initial parameter value. We use a variant of EM called *hard EM* which consists of the following steps:

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

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

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

In the steps above, MAP stands for *maximum a posteriori* and MLE is *maximum likelihood estimate*. As mentioned, multiple instances of this algorithm with random initialization are run which we do in parallel to boost performance as these instances are independent.

4.4.1 Parameter Initialization

The parameter p_c's for binary features are sampled from Dirichlet distribution Dir($[\alpha, \alpha]$) with being relatively large so they stay close to uniform. The initialization

distribution for $\pi_{c,i}$ is similarly 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_c's parameter 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.

4.4.2 E-Step

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

$$\log \mathcal{L}(Z, \boldsymbol{\theta}) = \sum_j \log \phi_j(Z_j) + \sum_{j,k} \log \phi_{\text{pair}}(Z_j, Z_k) \tag{4.8}$$

There is an Integer Program formulation of this problem by Kleinberg and Tardos called *Uniform Metric Labeling* [14]. For each node j (author in the graph) they 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 author-author graph we then have the potential $\log \phi_{\text{pair}}(Z_j, Z_k) = w_{A,A}(j,k)\left(d_{jk}\log(1-\tau) + (1-d_{jk})\log\tau\right)$ The Integer Program formulation can be relaxed to a Linear Program:

Minimize

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

Subject to

$$\sum_{c \in C} x_{j,c} = 1$$

$$d_{jk} = \frac{1}{2} \sum_{c \in C} d_{jkc}$$

$$d_{jkc} \geq x_{jc} - x_{kc}$$

$$d_{jkc} \geq x_{kc} - x_{jc}$$

$$x_{kc} \geq 0$$

This linear program can be solved by free or commercial software packages. We used Gurobi [10] for this purpose. Once the optimum x_{jc} are calculated, we picked the c with the highest x_{jc} as the cluster assignment for author j.

4.4.3 M-Step

In the M-Step we update model parameters θ with their MLE given the cluster assignments. The MLE estimates can be simply determined using frequency counts.

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

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

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.

$$\mu_c = \frac{\sum_{a \in C} l_a}{|c|} \tag{4.9}$$

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

4.5 The Method Overview

To recap, Algorithm 4.5 demonstrates how this clustering step fits into the overall spam classification method.

Algorithm 4.1 Review spam classifier

Input: Data set of reviews, their authors and the product being reviewed and a labeled set of spam reviews

Output: Set of spam reviews

1: Form the projected Author-Author graph $G_{A,A}$
2: Cluster $G_{A,A}$
3: Use the cluster assignment as feature
4: Train the classifier with a set of labeled spam reviews and a set of features

---

5. EXPERIMENTS

In this section we test our method. Initially, we verify that the proposed clustering method correctly recovers synthetically generated clusters. Next, we run the clustering on our Amazon reviews dataset to uncover clusters of reviewers with high collaboration. These cluster labels are then used as an additional feature in a review classifier to boost its performance. Finally, we compare our proposed author clustering method with another competing method for clustering in social networks based on SimRank similarity.

5.1 Evaluation on Synthetic Data

As a sanity check, we tested our method on synthetic data which is generated through the same process the model assumes. Later, we augment the dataset of reviews with the cluster assignments of the authors and re-train the SVM classifier to measure how much the author social context information can assist the classification

A synthetic collaboration graph of authors is generated. The generation process of the graph is described in Algorithm 5.1

Algorithm 5.1 Synthetic data generation procedure

Sample cluster sizes $\sim$ Dirichlet distribution
for each cluster c **do**
    Sample cluster parameters θ_c
    Sample μ (mean) of review log-length $\sim$ Gaussian (constant σ)
    for each of the three features described previously **do**
        Sample $p_c = \Pr(F_i = f | Z = c)$ uniformly.
    end for
    Sample cluster product preferences $\pi_c \sim$ Dirichlet distribution(α's $= 0.5$)
end for

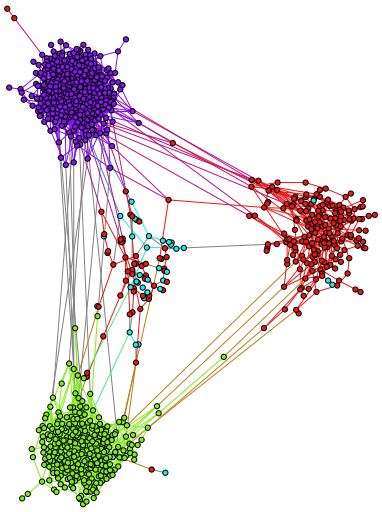

Figure 5.1: Synthetic clusters with the recovered clusters colored. There are 950 nodes and 2150 edges in the graph

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 akin to Figure 4.2. Finally the edges are sampled. The average degree is kept constant (30 in our 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 $1-\tau$. A resulting graph of such process with 4 clusters is shown in Figure 5.1. We run our method by over estimating the number of cluster 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 can be seen from Figure 5.1 the method can successfully recover most of the clusters as long as the intra-cluster edges occur more frequently than inter cluster edges. Most mistakes happen in the central cluster where the density is low. The

Table 5.1: Rand Index measure of the clustering on synthetic data

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

clustering method recovers 4 clusters which matches the true number of clusters and 94.3% of the nodes are clustered correctly. To obtain this value, we did as follows. Each recovered cluster is associated with the true cluster which the majority of its members belong to. All nodes whose detected cluster differs from the associated true cluster is considered as an incorrect clustering.

As another evaluation measure, we use *Rand Index* which is a measure of evaluating clustering when the ground truth is known. Graphs of size around 1.2K nodes and about 2.6K edges clustered into 4 clusters were generated using the process described earlier. Given different number of predetermined clusters to the clustering algorithm, we list the average Rand Index of 10 runs in Table 5.1.

One noticeable point in Table 5.1 is the improved clustering performance when the number of predetermined clusters highly 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 better chances of finding a more optimum final likelihood, hence better clustering.

5.2 Real World Dataset

The dataset was obtained by crawling Amazon product and author pages and extracting relevant information from those pages as described in Section 3. For this purpose, Scrapy was used [23] as the crawler framework. Additional details of obtaining our dataset is elaborated in Section 3.

Once the dataset was obtained, the Author-Author graph described in Section 4.2 was formed in which nodes are review authors and edges appear when a pair of authors write favorable reviews for the same products in the same time frame. This graph is shown in Figure 5.2. As a cleaning step, we discarded small connected components of this graph. That is, we only kept review authors who belonged to a connected component of size 10 or more. The size distribution of connected components in this graph is depicted in 5.3. The features of review authors are the ones described Section 4.3.1. Their mutual information is listed in Table 5.2 where the pair of reviews *No. Helpful + Not Helpful votes* and *Helpfulness Ration* have the highest mutual information. The value of mutual information suggests that the features used do not have significant marginal dependence.

We applied the described clustering method on the resulting author graph. EM algorithm was run with 16 random restarts the instances were let to continue until convergence. The predetermined number of clusters was set to 10. The value of τ determines how likely it is for connected nodes to be in the same cluster. That is a strength of this method where regardless of the number of pre-determined clusters, connected nodes with dissimilar clusters are punished. Hence the eventual number of emerged clusters can be less than what is predetermined. For high values 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 hence we considered them only. Inspecting the clusters shows us three distinct sets of users.

All listed detected clusters are dense (See Table 5.3). The people who posted deceptive reviews on products advertised on crowdsourcing websites are all contained in the first two clusters which is why we consider their reviews deceptive. The listed *shared characteristics* in Table 5.4 is based on a manual inspection of a sample of authors in each cluster as well as the most frequent terms in the product title of the

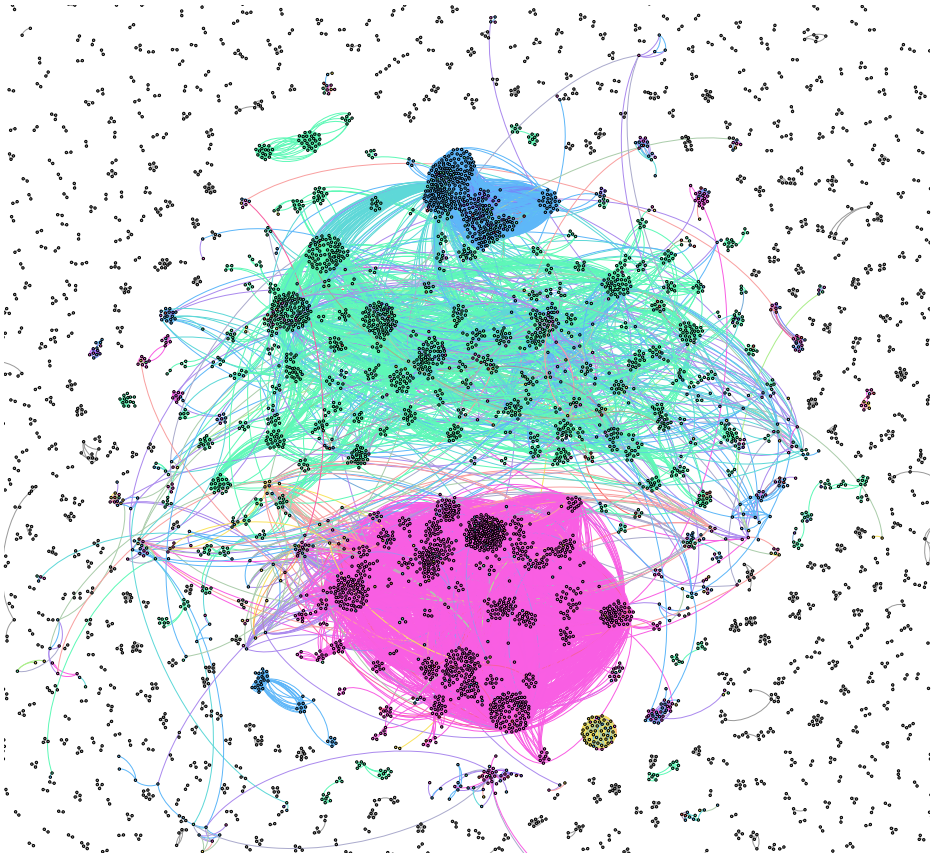

Figure 5.2: Color coded detected clusters in the graph of author collaboration. Each dot represents an author and each edge is a collaboration. The nodes and edges are colored based on the cluster they were assigned to

Table 5.2: Mutual information between pairs of features

	Review Length	Helpful Ratio	Helpful Total	Star Rating	Verified Purchase	hlpful fav unfav
Helpful Ratio	0.353					
Helpful Total	0.203	**1.726**				
Star Rating	0.029	0.1	0.04			
Verified Purchase	0.027	0.023	0.01	0		
hlpful fav unfav	0.023	0.026	0.02	0.001	0	
vrf prchs fav unfav	0.013	0.012	0.004	0.004	0.017	0.005

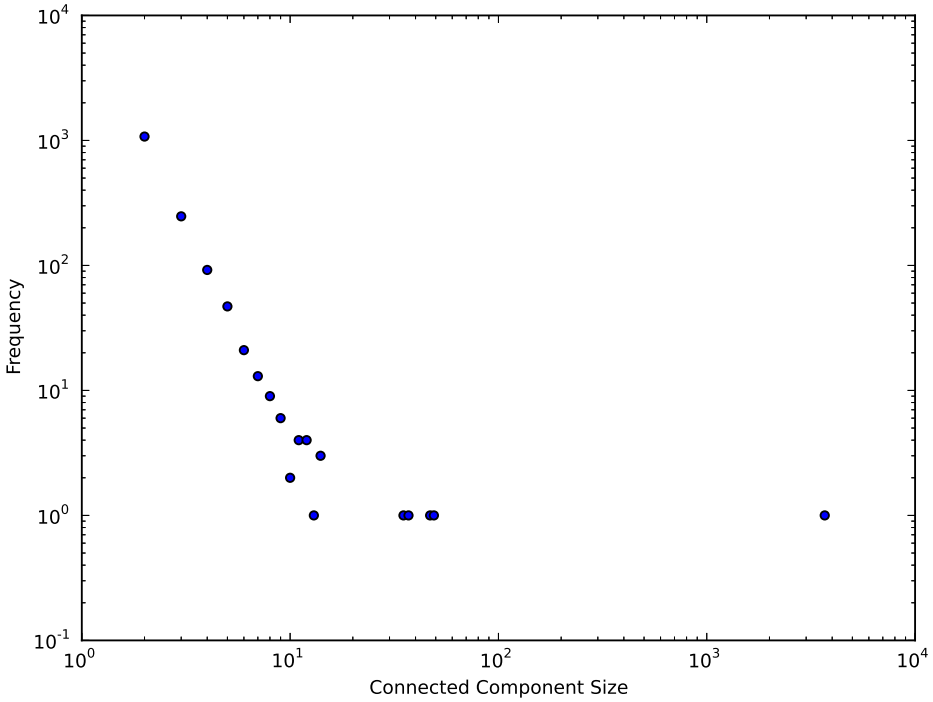

Figure 5.3: Log-log plot of the sizes of connected components of the Author-Author collaboration graph. The distribution follows the power-law

Table 5.3: Statistics for detected clusters of collaborating reviewers

Cluster	Size	Avg. Weighted Degree
C1	1079	14.5
C2	285	31.1
C3	1273	13.0
Everything Else	5217	2.3

Table 5.4: Identified groups of collaborating reviewers

Cluster	Characteristics	Real Name %	More Helpful on Favorable Reviews	More Verified Purchases on Favorable Reviews	Median Log-Review Length %
C1	Posted deceptive reviews for the same set of mostly health and beauty products	14%	14%	31%	5.5
C2	Posted deceptive reviews for a set of meditation books	40%	57%	42%	6.3
C3	Mostly users of Amazon Vine program who do not write deceptive reviews	45%	21%	8%	6.5

Table 5.5: Titles of top 20 products which were reviewed favorably by authors writing deceptive reviews. Sorted by the number of favorable reviews by deceptive authors

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 in Just 15 Minutes a Day.	Paperback
Vitamin D3	Health and Beauty
Tao Song and Tao Dance: Sacred Sound, Movement, and Power from the Source for Healing, Rejuvenation, Longevity, and Transformation of All Life (Soul 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	Kindle Edition
Tao II: The Way of Healing, Rejuvenation, Longevity, and I	Audible Audio Edition
The Journey: Captivity, Wilderness, Promised Land. Where are you Now? Where Will You Go?	Paperback
Best Age Spot Remover	Misc.
Peace of Mind: Healing of Broken Lives	Paperback
Diet Pills, Slimula Lose up to 20 Pounds in Just 4 Weeks!!! 60 Dietary Supplement, Slimming Capsules.	Health and Beauty
Melatonin	Health and Beauty

Table 5.6: KL-divergence between the distribution of the products rated by each cluster

	C1	C2	C3	S
C1	0.0	16.1	17.9	10.4
C2	16.6	0.0	15.9	13.3
C3	14.6	10.5	0.0	10.9

products reviewed. Table 5.5 lists a number of products associated with deceptive reviews. The last cluster is not composed of deceptive reviews, quite the opposite. Amazon has a program called *Amazon Vine* by which it gives the opportunity to a set of its top reviewers to review certain product before they become available on the website. This program results in the same behavior as that of deceptive reviewers, that is a large group of people reviewing similar sets of products in a short time window, many of them happen to be favorable. However they tend to write lengthier probably more elaborate reviews. Even though the non-spammer Amazon Vine participants form a cluster, they won't be considered as spam as the reviews they have written is not labeled so.

In modeling the clustering, each cluster of review authors review similar sets of products. This was modeled with a cluster preference distribution over products, i.e. chances of a member of the cluster reviewing each product. In Table 5.6 we have compared these distributions with the KL-divergence measure. Since this measure is not symmetric we have listed both values. $C1$ through $C3$ are the detected clusters described in Table 5.4. The set labeled with S is a random set of review authors the same size as that of the one it is being compared against.

5.3 Using Authors' Cluster Label to Classify Reviews

Here we have the same SVM classifier but augmented with the added feature of cluster labels of the authors. These cluster labels were obtained through the clustering method described above. As it can be seen from the ROC cureve, even though the classes are not balanced, knowing which cluster the review author belongs to significantly assists in the classification of reviews as spam/non spam (Figure 5.4). The area under the ROC curve (AUC) is bumped from 0.48 to 0.77. Also if we balance the classes, the AUC goes up to 0.95 compared to 0.86 for when cluster labels are not used.

The cluster labels which are categorical values (i.e. there is not an ordering for the values) are coded using vectors of indicator variables $(I_1, \ldots, I_k, \ldots, I_C)$ where $I_k = 1$ indicates the author of the review belongs to cluster k. Nevertheless representing clusters with a real number gave us similar performance. Such improvement in performance indicates the effectiveness of incorporating the social context of reviewers into deciding whether a review is spam or deceptive.

5.4 Clustering With SimRank and k-medoids

In this section we use an alternative method to cluster the review authors graph. It uses a similarity measure defined over graphs called SimRank [11]. Then a k-medoids clustering is performed based on the similarities or distances obtained from SimRank.

SimRank is a similarity measure between nodes of a graph. It is defined based on the premise that *two nodes are similar if they are related to similar nodes.* It is a recursive definition. The nodes of the graph can be heterogeneous. For instance, suppose a given graph has two types of nodes: One corresponding to text documents and another corresponding to textual tags of those documents. Then the SimRank

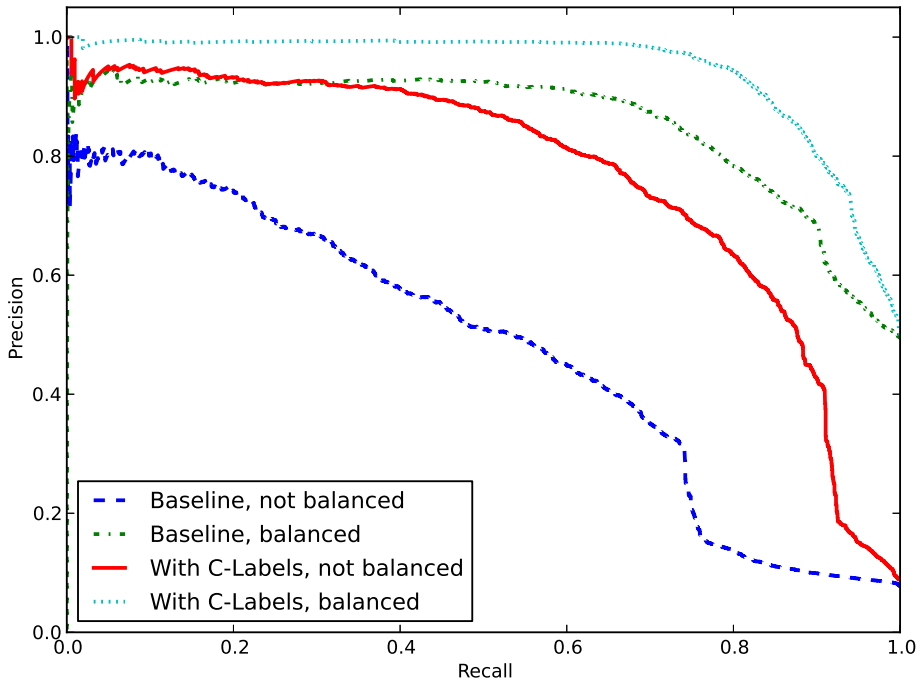

Figure 5.4: ROC curves of the performance of the review classification in four cases. The lowest performance belongs to the baseline SVM method (AUC = 0.48). The red line is the belongs to the performance when author clusters are used as an additional feature (AUC=0.77). The green dotted curve belongs to the performance of the baseline method when the class sizes are balanced with equal sizes (AUC=0.86). The last plot is the performance when the class sizes are balanced and the author cluster is used as a feature (AUC=0.95)

premise is that similar documents have similar tags and similar tags are applied to similar documents.

The calculation for SimRank is recursive and for an undirected graph its initial condition and recursive step are in equations 5.1 and 5.2. The value of $S_k(a, b)$ converges to the their SimRank score. In Equation 5.2 $0 < C < 1$ is called the *decay factor* and the symbol $\sim$ stands for adjacency in the graph.

$$S_0(a, b) = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases} \tag{5.1}$$

$$S_{k+1} = \frac{C}{|N(a)||N(b)|} \sum_{i \sim a} \sum_{j \sim b} S_k(i, j) \tag{5.2}$$

Initially we calculated SimRank score for the bipartite graph containing both review authors and products. This means two review authors are similar if they review similar products and two products are similar if they are reviewed by similar people. Once the similarity matrix is calculated. We use k-medoids to cluster the authors. k-medoids is a clustering algorithm similar to k-means but is useful when pairwise distances are known but the objects are not in a Euclidean space to calculate their mean. k-medoids cluster centers are initialized randomly.

Additionally, clustering with SimRank and k-medoids was performed on Author-Author collaboration graph $G_{A,A}$. The results of these two different clustering of review authors are compared to our proposed method.

Before discussing the performance, it is worth pointing few technicalities. For large data sets, there is a technique called *CLARA* in which initially a smaller sample of the databased is clustered and the resulting cluster means are used as initialization for the clustering on the original dataset. CLARA was used during k-medoids. The

steps are described in Algorithm 5.4.

Moreover, calculating SimRank is slow for large graphs. The original implementation time complexity is $O(n^4)$ where n is the number of the nodes. The memory complexity for the full graph of products and authors with over 95K nodes would require $\frac{95K^2 \times 64}{2} \approx 300GB$ of memory which is also not feasible. Therefore we performed the clustering on a subset of the original graph. The subgraph has 13.7 K authors and products. There are 14K reviews between them of which 4.7 K reviews are marked as deceptive. Our implementation of the SimRank uses MapReduce framework to parallelize the task. In order to speed up MapReduce, the delta-simrank method was used [2].

Algorithm 5.2 Clustering algorithm

1: **function** CLARA
2: sample ← small random sample from the data set
3: Run K-MEDOIDS(sample) multiple times with random restarts to cluster the sample
4: Use the resulting cluster centers as initial cluster centers for the k-medoids on the original dataset
5: **end function**
6: Run CLARA multiple times and return the clustering with least sum of square error

Once again, the cluster of each review author is used as a feature for the SVM based review classifier.

The first clustering which is over both authors and products performs poorly enough to be almost identical to the performance of the review classifier without it (Figure 5.5).

The same SimRank based clustering was performed on the Author-Author graph.

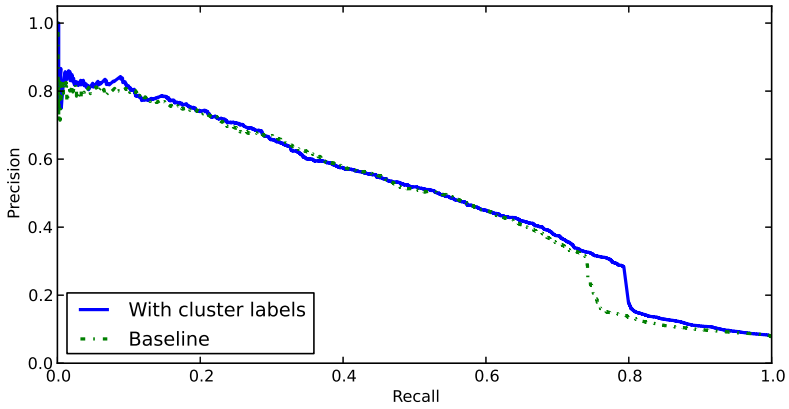

Figure 5.5: Comparison between the performance of the proposed method vs. using SimRank and similarity measure and k-medoids initiated using CLARA on a subgraph of review authors and products

Since this graph is small enough for the similarity matrix to fit into the memory ($8K$ nodes), we avoided the MapReduce overhead by running it on a single machine yet in parallel. We continued calculating SimRank until the changes in values were less than 0.001 which translated into around 8 to 10 iteration and takes several hours. The decay factor (C in Equation 5.2) was set to 0.8 following the suggestion from SimRank authors.

The results are shown in Figure 5.6. The performance of the SimRank based clustering is comparable yet slightly less than our proposed method. It should be noted that SimRank, at least the original version, does not take edge weights into account. Incorporating edge weights into a variant of SimRank could lead it to perform better. Still the main disadvantage of SimRank is its running time which has given rise to various techniques to approximate it with less computation cost which takes several hours compared to our proposed clustering which takes several minutes.

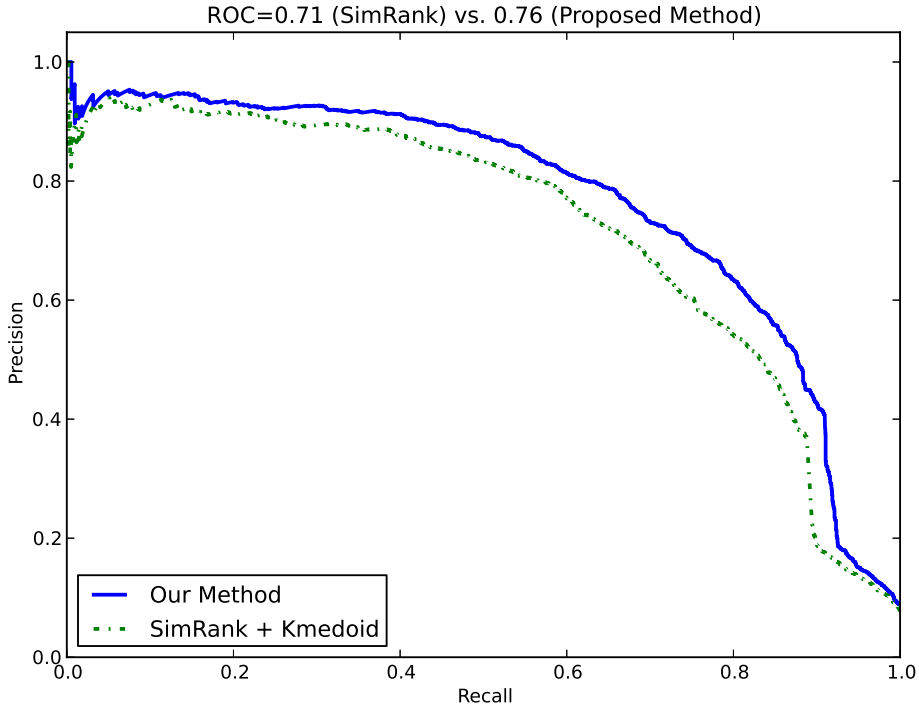

Figure 5.6: Comparison between the performance of the proposed method vs. using SimRank and similarity measure and k-medoids initiated using CLARA on author-author graph

6. CONCLUSION

We studied a case where users write favorable reviews for products on Amazon.com to get paid on crowdsourcing websites. We analyzed both the deceptive reviews and reviewers and noticed certain distinguishing characteristics of the reviews and the review authors. However those features were not distinguishing enough to allow for a reliable classification of reviews. Therefore, we used the observation in which review authors end up writing reviews for similar products. Based on this, we formed an author-author collaboration graph. We treated the graph as a Markov Random Field and performed the learning process using the EM framework. The results on the Amazon.com dataset allowed us to detect three prominent clusters of users two of which wrote deceptive reviews and the third one was users in Amazon.com Vine program which exhibits similar behaviors. We also verified the modeling on a synthetic dataset and showed how the utilizing the results of clustering can significantly improve the performance of a naïve deceptive review classifier. Additionally we implemented a competing method to cluster the review authors using SimRank similarity measure and k-medoids clustering. The resulting performance was slightly less than our proposed method while the computation cost was significantly larger.

There are however assumptions made during our study. First, the ground truth labels for products and users are based on the positive examples of products engaging in a deceptive review campaign and we assumed the rest of the crawled product have not solicited deceptive favorable review. However this is not guaranteed. Also while our clustering scheme has the advantage of not requiring to know the number of clusters in advance, there are certain assumptions made in the modeling. For instance, conditional independence of our features given a cluster. One way of im-

proving upon our method is to relax such assumptions. It will result in a model with higher number of parameter that needs more data to train.

REFERENCES

[1] Dragomir Anguelov, Daphne Koller, Hoi-Cheung Pang, Pen Srinivasan, and Sebastian Thrun. Recovering articulated object models from 3d range data. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 18–26. AUAI Press, 2004.

[2] Liangliang Cao, Brian Cho, Hyun Duk Kim, Zhen Li, Min-Hsuan Tsai, and Indranil Gupta. Delta-simrank computing on mapreduce. In *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, pages 28–35. ACM, 2012.

[3] Judith A Chevalier and Dina 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`.

[4] Cristian Danescu-Niculescu-Mizil, Gueorgi Kossinets, Jon Kleinberg, and Lillian Lee. How opinions are received by online communities: A case study on Amazon. com helpfulness votes. In *Proceedings of the 18th International Conference on World Wide Web*, pages 141–150. ACM, 2009.

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

[6] Christopher Elliott. Hotel reviews online: In bed with hope, half-truths and hype. `http://www.nytimes.com/2006/02/07/business/07guides.html` (Retrieved: 2013).

[7] Song Feng, Longfei Xing, Anupam Gogar, and Yejin Choi. Distributional footprints of deceptive product reviews. In *Proceedings of Sixth International AAAI Conference on Weblogs and Social Media*, pages 98–105, 2012.

[8] Chris Forman, Anindya Ghose, and Batia 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.

[9] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, pages 35–47. ACM, 2010.

[10] Gurobi Optimization Inc. Gurobi optimizer reference manual. `http://www.gurobi.com`, 2013.

[11] Glen Jeh and Jennifer Widom. Simrank: A measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*, pages 538–543. ACM, 2002.

[12] Nitin Jindal and Bing Liu. Opinion spam and analysis. In *Proceedings of the International Conference on Web Search and Web Data Mining*, pages 219–230. ACM, 2008.

[13] jobboy.com. `http://www.jobboy.com`, (Retrieved: 2012).

[14] Jon Kleinberg and Eva 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] Fangtao Li, Minlie Huang, Yi Yang, and Xiaoyan Zhu. Learning to identify review spam. In *Proceedings of the Twenty-Second International Joint Conference*

on Artificial Intelligence, volume 3, pages 2488–2493. AAAI Press, 2011.

[16] Yue Lu, Panayiotis Tsaparas, Alexandros Ntoulas, and Livia Polanyi. Exploiting social context for review quality prediction. In *Proceedings of the 19th International Conference on World Wide Web*, pages 691–700. ACM, 2010.

[17] microworker.com. `http://www.microworkers.com`, (Retrieved: 2012).

[18] Claire Cain Miller. Company settles case of reviews it faked. `http://www.nytimes.com/2009/07/15/technology/internet/15lift.html` (Retrieved: 2013).

[19] Arjun Mukherjee, Bing Liu, and Natalie Glance. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st International Conference on World Wide Web*, pages 191–200. ACM, 2012.

[20] Myle Ott, Claire Cardie, and Jeff Hancock. Estimating the prevalence of deception in online review communities. In *Proceedings of the 21st International Conference on World Wide Web*, pages 201–210. ACM, 2012.

[21] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 309–319, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[22] rapidworkwers.com. `http://www.rapidworkwers.com`, (Retrieved: 2012).

[23] scrapy web crawling framework. `http://scrapy.org`, (Retrieved: 2013).

[24] shorttask.com. `http://www.shorttask.com`, (Retrieved: 2012).

[25] Pawel Sobkowicz, Mike Thelwall, Kevan Buckley, Georgios Paltoglou, and Antoni 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.

[26] David Streitfeld. For $2 a star, an online retailer gets 5-star product reviews. `http://www.nytimes.com/2012/01/27/technology/for-2-a-star-a-retailer-gets-5-star-reviews.html` (Retrieved: 2013).

[27] David 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` (Retrieved: 2013).

[28] VIP deals rebate letter. `https://www.documentcloud.org/documents/286364-vip-deals.html` (Retrieved: 2013).

[29] Gang Wang, Christo Wilson, Xiaohan Zhao, Yibo Zhu, Manish Mohanlal, Haitao Zheng, and Ben Y Zhao. Serf and turf: Crowdturfing for fun and profit. In *Proceedings of the 21st International Conference on World Wide Web*, pages 679–688. ACM, 2012.

[30] Sihong Xie, Guan Wang, Shuyang Lin, and Philip S Yu. Review spam detection via temporal pattern discovery. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 823–831. ACM, 2012.