Spotting Spam Reviewers in Amazon Review Data

Siyuan Li siyuan.li@epfl.ch
Wanhao Zhou wanhao.zhou@epfl.ch
Yuanfei Mai yuanfei.mai@epfl.ch

Abstract

The widespread of the fake reviews in the online reviews system could be a problem for normal customers to make the right purchase decisions. Here we utilize the statistical model to study the behavioral patterns of spam reviewers and use clustering algorithms to distinguish the potential spam and normal reviewers. The case study on part of the Amazon Review Dataset shows that many potentially spam reviewers are detected by our methods among Amazon users.

1 Introduction

Nowadays, online reviews have increasingly become an important factor that can influence individuals when making purchase decisions, and this gives incentives for people to write fake reviews to promote or demote some products. These fake reviews can mislead customers to buy a product in poor quality or miss a good product. Given this problem, it is crucial to have an effective way to detect fake reviews and spam reviewers to alleviate the misleading effect of such reviews.

This problem can be seen as a classification problem. In the past few years, different methods based on supervised learning (Jindal et al., 2008; Lim, E.P. et al., 2010; Xie, S.et al., 2012) were put forward to spot these spam reviews or reviewers. Unfortunately, it is very difficult for people to obtain the training data by manually labelling reviews as fake or non-fake since many fake reviews can be bewildering enough to mislead customers.

Lim et al. (2010) reveals that the rating behaviors of spam reviewers are different from those of normal customers. Hence we believe that the data patterns of which deviate from the majority are the potential spam reviewers. In this paper, we aim to detect these spam reviewers and figure out the concrete abnormal behaviors of the spam reviewers. Afterwards, we can define some behavioral features to distinguish the fake reviewers from normal reviewers by unsupervised learning. In the case study part, we show our experimental results on Amazon Review Dataset.

2 Methodology

We formulate our methodology in two aspects, using statistical evidence and unsupervised learning algorithm respectively.

2.1 Abnormal Rating Patterns of Reviewers

We define the statistical unexpectedness to represent the deviation from the behavioral patterns of major customers (Jindal, N.et al. , 2010) and generate a ranking of customers.

2.1.1 Preconditions

For a review record, it contains attributes like *user id, product id*, etc., denoted as $\{\mathbf{A}_j\}_{j=1}^{m}$, where m stands for number of attributes. We classify the ratings into three classes: Positive (5 stars), Neutral (3 or 4 stars) and Negative (1 or 2 stars) and denote it as $\{c_i\}_{i=1}^{3}$. Given a specific data attribute, e.g. *user id*, we can obtain its distribution of ratings. Denote this rule as: $\mathbf{A}_j \rightarrow c_i$.

2.1.2 Unexpectedness

We suppose that most of the reviews are from normal customers, and the majority of reviewers are normal. Hence the unexpectedness represents the deviation from the normal customers. Furthermore, we hypothesize that attributes and ratings are statistical independent, i.e. $\mathbb{P}(c_i|\mathbf{A}_j) = \mathbb{P}(c_i)$.

The presence of spam reviewers leads to the unexpectedness, i.e. the deviation of $\mathbb{P}(c_i|\mathbf{A}_j)$ from $\mathbb{P}(c_i)$. Given the rule: $\mathbf{A}_j \rightarrow c_i$, we define the *Confidence Unexpectedness* (*CU*) and *Support Unexpectedness* (*SU*) based on conditional probability and joint probability respectively:

$$CU(a_{jk} \rightarrow c_i) = \frac{\mathbb{P}(c_i|a_{jk}) - \mathbb{E}[\mathbb{P}(c_i|a_{jk})]}{\mathbb{E}[\mathbb{P}(c_i|a_{jk})]}$$

$$SU(a_{jk} \to c_i) = \frac{\mathbb{P}(c_i, a_{jk}) - \mathbb{E}[\mathbb{P}(c_i, a_{jk})]}{\mathbb{E}[\mathbb{P}(c_i, a_{jk})]}$$

where a_{jk} is a possible value of the attribute $\mathbf{A}_j$.

In our case, $\mathbf{A}_j$ refers to *user id*. We compute the *CU* and *SU* for each *user id* in each class. The sum of the *CU* and *SU* in three rating classes for each user is used to rank in descent order to filter top ones as potential spam reviewers, i.e. $CU(a_{jk}) = \sum_{i=1}^{3} CU(a_{jk} \to c_i), SU(a_{jk}) = \sum_{i=1}^{3} SU(a_{jk} \to c_i)$ Theoretically, the *CU* and *SU* rankings could spot the suspect users who are inclined to give high or low ratings.

2.2 Reviewer and Review Clustering

In the above method, we obtain some behaviors of potential spam reviewers. In addition, according to Mukherjee et al. (2013), we define other features for our unsupervised clustering algorithm.

2.2.1 Reviewer Features

a. Review Text Similarity. Fake reviewers tend to copy or modify a little based on their previous reviews to save time. Hence we can compute the maximal text similarity f_a for each reviewer.

$$f_a(u) = \max_{r_i, r_j \in \mathbf{R}_u, i \neq j} \cos\left(E(r_i), E(r_j)\right)$$

where u is a reviewer, $E(\cdot)$ is sentence embedding function, and $\mathbf{R}_u$ refers to all reviews by u.

b. Maximal Number of Reviews Per Day. Spam reviewers may write a lot of reviews each day to earn money. Thus, we compute the maximal number of reviews per day for each reviewer ($\mathrm{MNR}(u)$) and normalize it by the maximal number of reviews among all the reviewers.

$$f_b(u) = \frac{\mathrm{MNR}(u)}{\max_u(\mathrm{MNR}(u))}$$

c. Review Interval. The interval between the first review date $F(u)$ and the latest review date $L(u)$ of the same reviewer could be short for spam reviewers since they need to frequently write reviews (Mukherjee, A.et al. , 2012). If all reviews are posted within a short time ($\tau = 28$ days), the reviewer is likely to be a spammer.

$$f_c(u) = \max(1 - \frac{L(u) - F(u)}{\tau}, 0)$$

d. First Reviewer Frequency. If a spam reviewer is hired to promote or demote a certain product, he / she will strive to be the first reviewer under this product due to the herding effect (Lim, E.P. et al., 2010). We compute the corresponding frequency of being the first reviewer.

$$f_d(u) = |\{r \in \mathbf{R}_u : r \text{ is the first review}\}|/|\mathbf{R}_u|$$

The four features above will be in the range of $[0, 1]$, and values close to 1 imply a higher probability of being a spam reviewer.

2.2.2 Review Features

e. Review Repetition. Fake reviewers may work in a group thus there would be similar reviews under a particular product (Lim, E.P. et al., 2010). We compute the review similarity (after sentence embedding) under each product.

$$f_e(u, p) = \mathbb{1}_{\{\exists r \in \mathbf{R}_p:\ \cos\left(E(r_{u,p}), E(r)\right) \geq \beta\}}$$

where $r_{u,p}$ is the review by user u under product p, $E(\cdot)$ is sentence embedding function, and $\mathbf{R}_p$ denotes all the reviews under product p. We choose $\beta = 0.9$. Given u and p, $r_{u,p}$ is unique.

f. Rating Bias. Whatever the quality of products, the spam reviewers tend to give positive or negative ratings to promote or demote the products. This leads to the bias of the spam reviewer ratings compared to the average ratings of normal customers.

$$f_f(u, p) = \mathbb{1}_{\{c_{u,p} \notin [q(p)_\alpha,\ q(p)_{1-\alpha}]\}}$$

where $c_{u,p}$ is the rating by user u under product p, and $q(p)$ is the quantile function of the rating distribution of product p. We choose $\alpha = 0.1$.

g. Extreme Rating. Spam reviewers tend to give extreme ratings. In function $f_g(u, p)$, output 1 indicates an extreme rating.

$$f_g(u, p) = \mathbb{1}_{\{c_{u,p} \in \{1,5\}\}}$$

h. Early Post. A spam reviewer tends to be the first reviewer under a product, so we define the early time frame to represent this behavior. If a review comes in the first 3 days compared with the first review, then it might be a fake review (Lim, E.P. et al., 2010). The output f_h is either 0 or 1.

The review features above take value in $\{0, 1\}$, corresponding to non-fake reviews and fake reviews respectively.

2.2.3 Unsupervised Clustering

We have generated 4 features for each reviewer and review respectively. For each reviewer and review, we use *K-means* and *Gaussian Mixture Model (GMM)* to divide them into two clusters.

3 Case Study: Amazon Review Dataset

We use Amazon Review Dataset (McAuley, J.et al. , 2015) described in Table 1. We experiment on the category: *Home and Kitchen*.

# Reviews	# Products	# Reviewers
551,682	436,988	66,519

Table 1: Description of the Dataset

3.1 Method 1: Results and Analysis

We consider the attribute *user id* and the corresponding rating behaviors to calculate *CU* and *SU*.

3.1.1 Confidence Unexpectedness (*CU*)

We first compute the $CU(a_{jk})$ for each user, which quantifies the difference between rating distribution of a specific user and the average rating distribution of all users. Larger $CU(a_{jk})$ means the rating behavior of this particular user deviates more from that of average ones. We sorted the user list based on *CU* and regard the top 10% users as suspicious. Figure 1 shows the rating distribution of the most suspicious user. For the full list of suspicious users, please refer to our notebook.

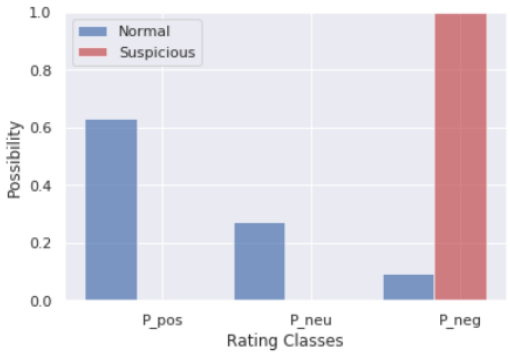

Figure 1: Rating Distribution of a Spam Reviewer vs. Average Distribution of All Users

3.1.2 Support Unexpectedness (*SU*)

Table 2 shows the result sorted by *CU*. We can see that many users have the same *CU* values and we cannot rank them only based on *CU* values. Therefore, we introduce the Support Unexpectedness. Intuitively, we want to quantify the difference between those users with the same *CU* value by assigning weights to them based on the number of reviews given by them. If two users only wrote negative reviews, i.e. having the same *CU*, the one wrote more reviews is more suspicious.

The results of *SU* show that reviewers who write more reviews would be placed at a higher rank given the same *CU*.

reiviewerID	*CU*	*SU*
A1LUPQJLN3BRDO	7.734	12.5375
A2UW6M4S2W8QOC	7.734	12.5375
A13HOP8991UOLC	7.734	6.0635
A23YJ2MIV0EKI6	7.734	6.0635
A1KJM9PS3JJH8P	7.734	4.7687

Table 2: Top Suspicious Spam Reviewers Ranking Based on *CU* and *SU*

3.1.3 Analysis

We present the results in scatter plots, shown in Figure 2 and 3, with three axes representing the values of positive, neutral and negative unexpectedness respectively. We can clearly observe the separation hyperplane between spam (red points) and non-spam (blue points) reviewers.

Using the metric defined in Section 2.1.2, we rank the users using $\{CU, SU\}$. Top users have the similar pattern: most of their ratings are neutral or negative. This is reasonable since the majority of ratings in our dataset are positive, and thus unexpectedness arises from giving too many non-positive ratings.

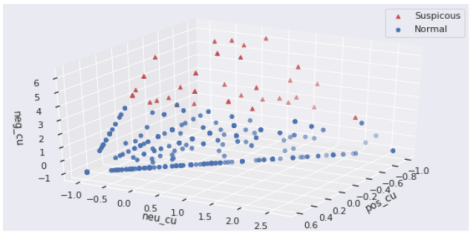

Figure 2: Illustration of *CU*, Sample 10% of Users

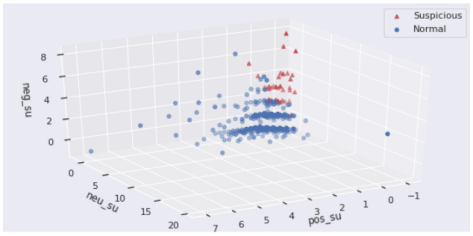

Figure 3: Illustration of *SU*, Sample 10% of Users

The drawback of this approach is clear. Since the positive ratings account for a huge proportion (83%), even users who give all positive ratings will still be considered as normal by Method 1. Furthermore, the review text given by reviewers should be an important factor to be considered as well. Thus, we experiment with Method 2.

3.2 Method 2: Results and Analysis

After computing the features for reviewers and reviews defined in Section 2.2 respectively, we normalize the data and use K-means and Gaussian Mixture Model to cluster reviewers and reviews.

3.2.1 Reviewer Clustering

The numbers of suspicious spam and non-spam reviewers detected by K-means and GMM are shown in the Table 3.

Algorithm	K-means	GMM
# Suspicious Reviewers	4,685	5,831
Suspicious Reviewers (%)	7.0%	8.8%

Table 3: The Numbers of Suspicious Spam Reviewers Found by K-means and GMM

For GMM, we plot the histogram (Figure 4) for the difference between the probabilities of two classes. The histogram implies that there is a clear separation between two classes.

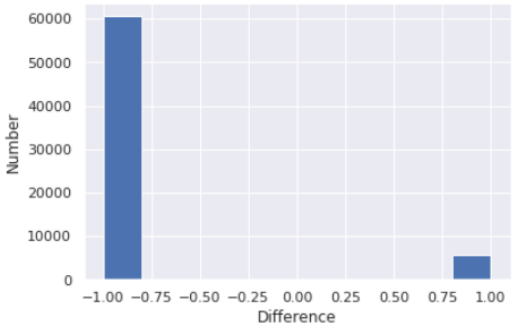

Figure 4: Histogram for the Difference between the Probabilities of Two Classes

3.2.2 Review Clustering

The clustering results of fake and non-fake reviews are presented in the Table 4. Since features are binary, we only use K-means.

Algorithm	K-means
# Suspicious Reviews	213,590
Suspicious Reviews (%)	38.7%

Table 4: The Number of Suspicious Spam Reviews Found by K-means

Hereafter we trace back the suspicious spam reviewers from the detected fake reviews. We calculate the proportion of the fake reviews for each reviewer. If the proportion exceeds 80%, we regard the reviewer as spam.

Consequently, we obtain another list of spam reviewers. We calculate the intersection of two lists. There are 522 users discovered by two clustering methods in common, and the percentage of overlapping between two lists is around 10.0%.

3.2.3 Analysis

Since the intersection between two methods is small, we visualize the histogram of percentage of fake reviews for each reviewer in Figure 5. We find that the proportion of users with at least one detected fake review is huge: 94.6%. Intuitively, the ideal histogram of the percentage of the fake reviews for reviewers should be descending as the percentage goes from 0 to 1. However, contradictory figure implies that the clustering of reviews is unconvincing.

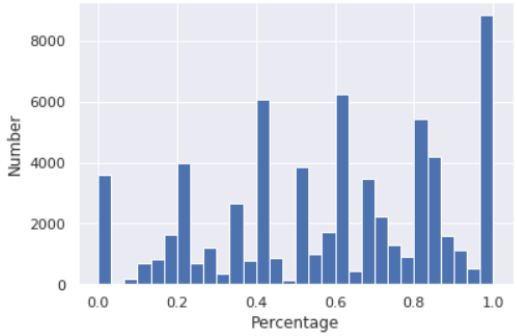

Figure 5: Histogram of Percentage of Fake Reviews for Reviewers

We tried to tune our review clustering model with different thresholds for features but it made little difference. After close observation of the data, we see that the problem arises from two aspects. First, it is indeed hard to tell whether a review is fake or not even by human inspection. Also, the designed features might be problematic. For instance, the Early Post feature is more reasonable when compared with the launch day of a product, but the relevant data is missing in our dataset.

4 Conclusions

In this paper, we proposed two methods, studying the abnormal rating patterns of reviewers and unsupervised clustering, to spot spam reviewers. Our spam detection model performs the spam reviewer classification in the absence of manually labeled data. In our experiment, we report a case study using reviews on Amazon Dataset, where we found many suspicious reviewers. The result of reviewer clustering is promising.

References

Feng, S., Xing, L., Gogar, A. and Choi, Y. 2012. *Distributional Footprints of Deceptive Product Reviews* ICWSM, 12, pp.98-105.

He, R. and McAuley, J. 2016, April. *Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering.* In proceedings of the 25th international conference on world wide web (pp. 507-517). International World Wide Web Conferences Steering Committee.

Jindal, Nitin and Liu, Bing 2008. Alternation. *Proceedings of the 2008 international conference on web search and data mining*, (pp. 219–230.) ACM.

Jindal, N., Liu, B. and Lim, E.P. 2010, October. *Finding unusual review patterns using unexpected rules*. In Proceedings of the 19th ACM international conference on Information and knowledge management (pp. 1549-1552). ACM.

Lim, E.P., Nguyen, V.A., Jindal, N., Liu, B. and Lauw, H.W. 2010, October. *Detecting product review spammers using rating behaviors.* In Proceedings of the 19th ACM international conference on Information and knowledge management. (pp. 939-948). ACM.

McAuley, J., Targett, C., Shi, Q. and Van Den Hengel, A. 2015, August. *Image-based recommendations on styles and substitutes*. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 43-52). ACM.

Mukherjee, A., Liu, B. and Glance, N. 2012, April. *Spotting fake reviewer groups in consumer reviews*. In Proceedings of the 21st international conference on World Wide Web (pp. 191-200). ACM.

Mukherjee, A., Kumar, A., Liu, B., Wang, J., Hsu, M., Castellanos, M. and Ghosh, R 2013, August. *Spotting opinion spammers using behavioral footprints*. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 632-640). ACM.

Xie, S., Wang, G., Lin, S. and Yu, P.S. 2012, August. *Review spam detection via temporal pattern discovery*. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. (pp. 823-831). ACM.