CSCE 5014 - Graph Theory
Final Project Report
on
Mitigating Bias in Online Review By Identifying Malicious Reviewers

Md Jubaer Hossain Pantho
Department of Computer Science and Computer Engieeneering
University of Arkansas
Student ID: 010771209
mpantho@uark.edu

May 7, 2018

1 Abstract

In this era of internet, online reviews provide a valuable resource for potential customers to make the right choice of products. Nevertheless, the large variations in the review quality present a big impediment to the effective use of the reviews. Sometimes there exist malicious reviewers who attempt to disrupt the soundness of certain products review. Automatically identifying these users is a challenging task. This work presents a model to mitigate bias in online review by automatically detecting untrusted reviewers and excluding them from the calculation. Our empirical study on the Amazon food reviews dataset demonstrates that the proposed approach is highly effective.

2 Introduction

The increasing impact of the Internet has dramatically changed the way that people shop for goods. Consumers are surrounded by online reviews thanks to other consumers who've gone to the trouble of posting opinions about products and services online. Such reviews have become an indispensable component of e-commerce Websites such as Amazon, and they are also available through dedicated Websites such as CNET and IMDB.

These days, reputations are almost never a blank slate. While reading reviews can help the potential customers make informed decisions, in many cases the reviews available for a product may contain bias. The sheer volume of reviews can be overwhelming and actually impede the customers' ability to evaluate the product. Moreover, They're reviews posted by malicious reviewers which are designed specifically to give a false impression to consumers on the point of purchasing. It is important to single out this reviewer to ensure reliable reviews for consumers. Different websites applies various ways to alleviate this problem. For instance, Amazon provides a verified purchase tag on different reviews [2]. In this case, review request is sent directly to the consumer after purchase is authorized by that consumer. However, this does not provide a concrete solution for products with small number of reviews. Many websites are now allowing readers of a review to indicate whether they think that review is helpful by voting for or against it, and a tally (or score) is provided in the form of "100 out of 150 people found the following review helpful". The reviews can be sorted according to their helpfulness using those scores [3]. Although this is certainly an improvement, there are still important issues to be addressed. For example:

- For newly posted reviews, most likely no vote or only a few votes have been cast, and therefore, identifying their helpfulness is difficult.
- Presenting the reviews ranked by their user-voted helpfulness scores may create situations of "monopoly" in that only the highest ranked reviews get viewed, leaving no opportunities for the newly published yet unvoted reviews to show up on users' radar.
- There are consumers who buy product based on overall rating of the product. Since these system does not modify the overall review, they are seldom helpful to these kind of consumers.
- In some cases, reviews can be incorrectly labeled as helpful or not helpful due to spam voting.

In these scenarios, it can be beneficial to single out malicious reviewers instead of identifying malicious reviews. By keeping these biased out of the equation we can minimize bias in overall product review for different products.

The main contributions of this work are:

- Automatically identify malicious reviewers in online review system. This is done by combining multiple methods. i.e. Observing variance in review rating for different users, examining timestamps, review summary.
- Provide an updated overall rating for different products to help consumers make better choice.

The rest of the report is organized as follows: In Section 3 we discuss some background of this work. Section 4 and 5 describes the used dataset and general notions of our model. In section 6 we detail our proposed model. Section 7 presents the results we obtained with our model.

3 Background

This section starts by discussing work on aggregate rating that exist in the literature. Then we argue the behavior of biased reviewers.

In "Star Quality: Aggregating Reviews to Rank Products and Merchants", M. McGlohon et. al. cite a Wall Street Journal article that notes the "average rating for top review sites is a huge 4.3 out of 5 stars,"[4]. The authors use nine separate

techniques to produce a more accurate aggregate score with data from Google Product Search. These include median reviews, lower bound of a normal and binomial confidence interval, percentile of order statistic across websites, and filtering for anonymous or spam detection. One issue they find is that they could not easily test the accuracy of their algorithms because a product's true quality is not known. To work around this, they develop a method of making a test set by taking pairs of reviews from the same author and see if their methods can choose which product received a higher rating. Using this, they find that none of their algorithms were any more powerful than simply taking the average aggregate rating; the average aggregate ranking is 70% accurate across all 3 datasets, with the lower bound of the normal confidence interval replicating those results, and all other methods slightly worse.

This work motivates our problem, and emphasize the importance of mitigating bias in online review. Given our data we evaluated fake or biased reviewers based on the following criteria:

- **Consider the length and tone of the review.** If all the reviews are same and very short, it may be a fake.
- **Beware if the person has submitted a lot of reviews in a short period.** If a reviewer is being paid to write reviews, they may have written a great number of reviews on the same day.
- **Polyamorous reviewer:** These reviewers give every product a glowing review.
- **Always unsatisfied reviewer:** These reviewers give low score to all products. This is also not acceptable.

By removing these reviewers from the graph, we hope to minimize bias in the aggregate rating of products.

4 Data Set Information

We use an Amazon product dataset (Web data: Amazon Fine Foods reviews) compiled by Jure Leskovec at Stanford University [1]. This dataset consists of reviews of fine foods from amazon. The data span a period of more than 10 years, including all 500,000 reviews up to October 2012. Reviews include product and

user information, ratings, and a plaintext review. The Dataset Statistics is given in Figure 1.

Dataset statistics	
Number of reviews	568,454
Number of users	256,059
Number of products	74,258
Users with > 50 reviews	260
Median no. of words per review	56
Timespan	Oct 1999 - Oct 2012

Figure 1: Data Statistics

```
product/productId: B001E4KFG0
review/userId: A3SGXH7AUHU8GW
review/profileName: delmartian
review/helpfulness: 1/1
review/score: 5.0
review/time: 1303862400
review/summary: Good Quality Dog Food
review/text: I have bought several of the Vitality canned dog food products and have
found them all to be of good quality. The product looks more like a stew than a
processed meat and it smells better. My Labrador is finicky and she appreciates this
product better than most.
```

Figure 2: Data Information

The data format of this data set is illustrated in Figure 2. For each review we have kept the following information to generate the graph:

- Reviewer ID
- Product ID
- Review Score
- Review Time
- Review Summary

We further restricted our data to ensure that we had sufficient richness per reviewer to test our model while being cognizant of computational restraints. Our final dataset contained data from approximately 20,000 reviewers.

5 Graph Structure and General Notions

The general structure of our network is an undirected heterogeneous graph $G = (V, E)$. There are two type of nodes(V) in the network denoting user node V_u and product node V_p. We use a node attribute X_t to differentiate these nodes. Besides, all nodes have another attribute, X_{eval}. For user nodes V_u this attribute denotes trustworthiness. Whereas, for product node V_p this attributes holds the overall modified rating of the product. A review of a product from a reviewer will denote an edge (E) in the graph (reviewer node to product node). This edges has several attributes (Y) listed in the previous section. We generated the X_{eval} node attribute using our model. We analyze the outgoing edges attribute of a user node to certify the reviewer as a trusted reviewer or not. Figure 3 explains the structure of our network:

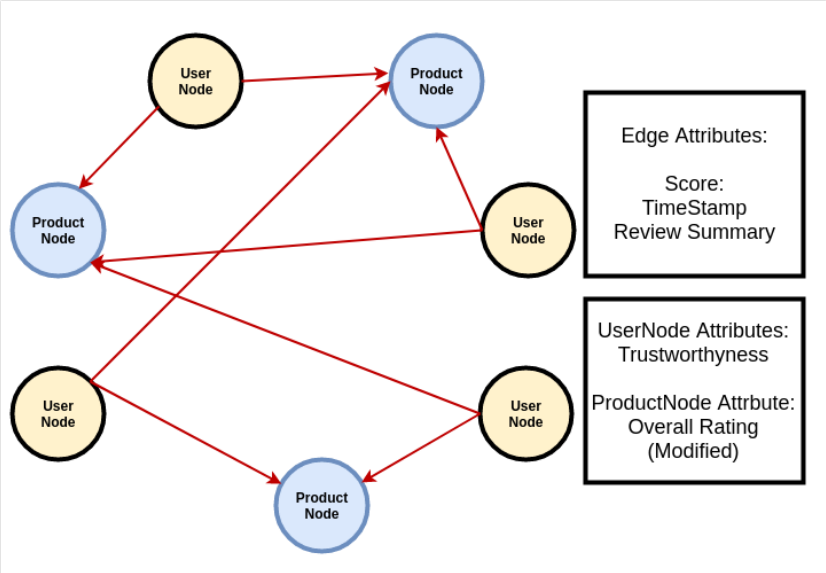

Figure 3: Network Structure with Product and User Node

6 Proposed Model

In this section, we explained our approach to identify biased reviewers. Later, we describe our procedure on updating the aggregate rating of different products.

6.1 Identify Bias

In section 3, we discussed the traits of fake or unreliable users. Our model iterates through each user node and attempts to identify suspicious behavior from the node attributes. Since users with only a few product review have minimum impact on the overall system our model opt out infrequent users from the calculation. If a user node has more than 20 outgoing edges then we consider this node as a node of interest. We examine the review score attribute of all the outgoing edges of a node and calculated the variance of score. If the variance is very low then we can assume that this is a polyamorous reviewer and thus not to be trusted.

Since fake reviewers tend to submit the same review for multiple products we examined the similarity of different reviews of a user by performing a text matching on review summary. We classify an user untrustworthy if we detect large number of similar reviews over multiple products.

Reviewers who are being paid to submit reviews usually submits a great many number of reviews on the same day. It is important to single our these users since they inflict maximum damage on the overall review system. We identify these type of reviewers by performing a timestamp check on time attribute over all outgoing edges of a user node. If a reviewer submits more than a certain number of reviews on a single day we tag this node as untrustworthy.

We believe that by combining these three criteria in our model we can effectively classify trusted users from the biased ones.

6.2 Updating Aggregate Rating

Once we tag all the reviewer nodes, the system iterates through all the product node and calculates the overall rating of the product by analyzing the incoming edge attributes. However, if an edge is coming from an untrustworthy node we exclude that edge from the calculation. The overall rating is then measure by taking simply the average over all the scores of that product [4].

7 Implementation and Results

In this section we present our implementation details and results obtained by evaluating our dataset.

Given the nature of the dataset we presumed that infrequent users have

minimum impact on the overall rating. Besides, it is tough to understand the integrity of a user from one or two reviews. As a result, in our final calculation we excluded the users with small number of reviews. We analyzed the dataset to identify the density of infrequent users. In order to do this we randomly selected 19777 users and calculated the number of reviews they submitted. Table 1 illustrates the results.

Table 1: Dataset Analysis of Reviewers with Different Number of Reviews per Head (total user: 19777)

No. of Reviews Submitted	**No. of Reviewers**	**No. of Reviews Submitted**	**No. of Reviewers**
1	7120	11	265
2	2804	12	218
3	1476	13	195
4	1578	14	260
5	1201	15	193
6	917	16	133
7	580	17	102
8	463	18	103
9	671	19	84
10	348	>=20	1066

Figure 4 provides a pictorial view of this result. As we can see we have less number of reviewers who have submitted a lot of reviews. This is expected. Whereas we have a large volume of reviewers who only submitted single or two reviews. Please check the source file to view full profiled data of users (userdata.txt).

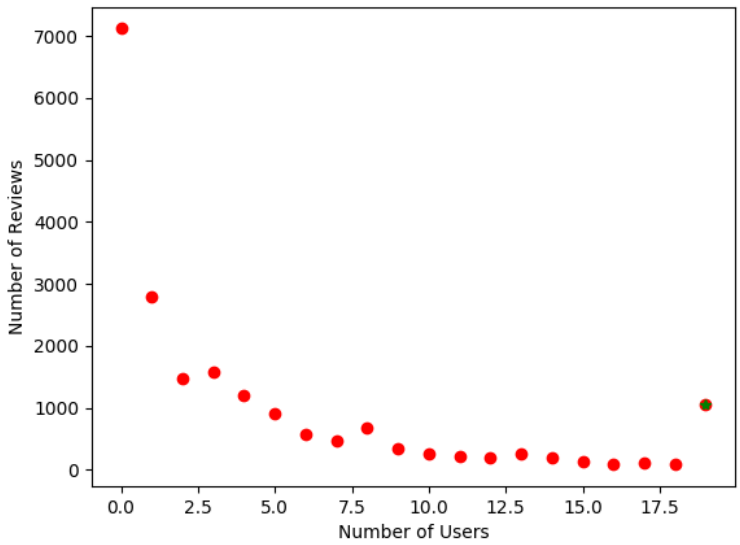

Figure 4: Number of Reviews vs Number of Users Submitting That Many Reviews

In order to understand the effectiveness of taking variance of rating as a measure of calculating trustworthiness we randomly selected some reviewers data with more than 20 reviews and plotted them on graph. Figure 3 illustrates the rating pattern of 9 of those users. As it is seen from the right-top image the behaviour of this user is somewhat unusual from the rest of them. This user gave 5 stars to almost all the products. Thus we will get a low variance score compared to the rest of the users and we can assume that this is a biased user.

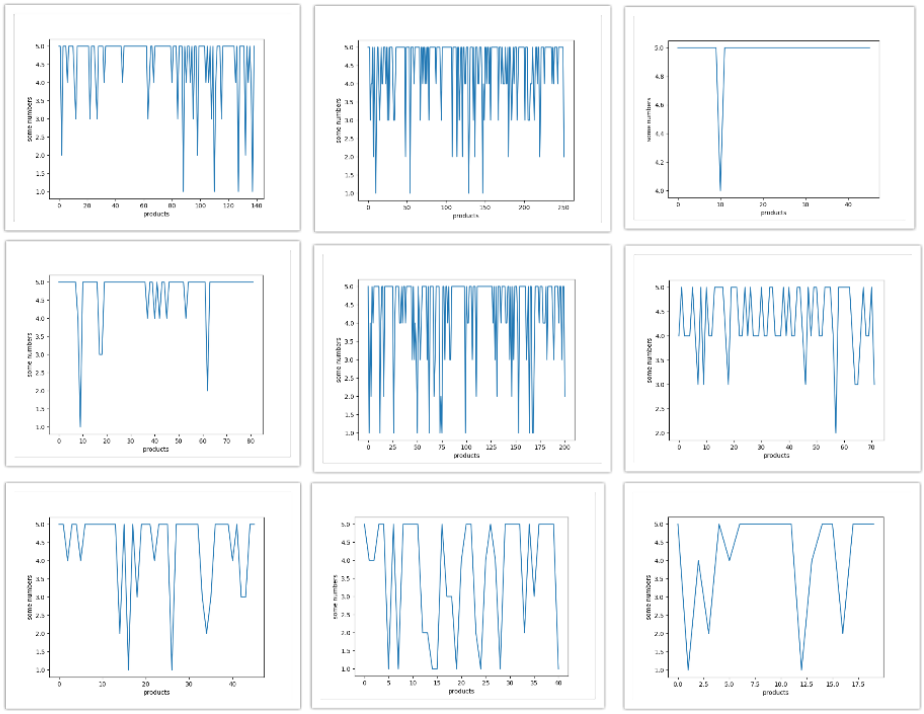

Figure 5: Variance analysis in Review for different reviewers.

One of the problem we faced in this calculation is how do we define the threshold for low variance. To get around this problem, we propose to calculate the mean of all the users' variances in the dataset. If user in question shows a variance lower than this mean variance we tag this user as non-trusted. However, in our current implementation we found that taking a variance threshold value of 0.1 provides good results.

Besides we calculated the timestamp differences of reviews for each user. The dataset provides a timestamp of each review in unix time. If a user has too many reviews within a small time range we tag the user node as non-trusted. Please look at the comments provided in the source code for more detail of this implementation.

Lastly we performed a text check to identify users who submits the same review for multiple products. Because chances are these reviewers provide little information in their reivew. The model tags an user non-trusted if the user submits the same review summary in one third of the time.

At this point, it needs to be mentioned that these three approaches uses threshold values to identify non-trusted user. By taking a lower threshold value we can increases the number of trusted reviewers. In contrast, higher threshold values will yield small number of trusted reviewers.

After classifying trusted and non-trusted users, the model updates the overall rating of products by itering through each product node and taking mean of edge scores. While calculating mean the model only considers incoming edges from trusted users.

We tested our prototype on 200 frequent users (with more than 20 reviews) selected randomly from the previously sampled data set of 20000 users. These users submitted reviews on 4500 products. While updating the overall score of the products we only consider products those have multiple reviews from these reviewers. Figures 6 shows some sample results of different trusted and non-trusted users classified by our model.

Figure 6: Sample results generated by our network classifying trusted and non-trusted users. Left column showing data on trusted users. Right column illustrates non-trusted users. See the attached **results.txt** file for detailed results on all users and products.

As you can see, non-trusted users exibit low variance and multiple submissions within a short period of time. Table 2 summarizes the result.

Table 2: Classification of Biased Reviewers (Dataset of 200 reivewers)

Criteria	Biased Reviewers	Trusted Reviers
Based on variance (threshold=0.1)	17	183
Based on number of multiple similar Review (threshold=33%)	11	189
Based on number of multiple reviews in 48 hours	15	185
Combination of above three (our model)	32	168

As we can see from the Table 2, it is possible to indentify more biased reivewers when we combine the above mentioned three criteria. Table 7 shows the modified updated rating for different products. We observed that for certain product there were major modifications.

```
product id : B000G6RYNE : 4.12903225806 / 4.02380952381
product id : B000G6MBX2 : 4.6 / 4.33333333333
product id : B000G6O2QG : 4.12903225806 / 4.02380952381
product id : B001LG940E : 4.0 / 3.8
product id : B004SKITAG : 4.0 / 3.0
product id : B003JA5KLM : 4.2 / 4.25
product id : B004LL97EO : 4.45454545455 / 4.375
product id : B007PA32OE : 4.2 / 4.25
product id : B000GWKO7G : 4.6 / 4.33333333333
product id : B00472I5A4 : 4.12903225806 / 4.02380952381
product id : B000LKXBL4 : 4.12903225806 / 4.02380952381
product id : B0029K68ZK : 4.5 / 5.0
product id : B000HDK0DC : 4.5 / 5.0
product id : B003Z6ZGZU : 3.5 / 2.0
```

Figure 7: Updated aggregate rating of different products generated by our network. Data format is : (normal rating)/(modified rating). In modified rating non-trusted users were excluded.

8 Conclusion

In this work, we present a model to classify non-trusted reviewers in online review systems by describing the problem as a graph structured problem. Cor-

recting bias in online reviews can sway important decisions in our economy. Our research offers hope for making online reviews more useful by reducing bias. There are a number of directions in which this research can be taken. We can further improve this model by incorporating techniques on detecting biased edges. Exist models can be combined (helpfulness/verified purchase) with our model to ensure better consumer experience.

9 Appendix: Source Files Explanation

The source files of our model are located at *source-files* directory. The file **true-review.py** contains the implementation of our model. To run this code execute the following command:

[python2 true-review.py > result.txt]

Please make sure to keep data-set file (**foods-100-reviewer.txt**) on the same folder. The file **user-data.txt** holds the information of user id and the number of reviews they submitted. Since the original data-set is very huge it is excluded from the submission file. The link for the original file is given in [1]. The generated results are stored in **results.txt**. We used the code **data-sample-generation.py** to generate our sample dataset. Please look at the comments provided in the code.

10 Reference

1. Jure Leskovec. Web data: Amazon Fine Foods reviews. Stanford University (access time: April, 2018)[source: $https://snap.stanford.edu/data/web-FineFoods.html$]

2. Amazon Verified Purchase Review.[source: $https://www.amazon.com$]

3. Li, Mengxiang & Huang, Liqiang & Tan, Chuan-Hoo & Wei, Kwok-Kee. (2013). Helpfulness of Online Product Reviews as Seen by Consumers: Source and Content Features. International Journal of Electronic Commerce.

4. McGlohon, M., Glance, N., Reiter, Z. 2010. Star Quality: Aggregating Reviews to Rank Products and Merchants, in: Proceedings of Fourth International Conference on Weblogs and Social Media (ICWSM), pp. 114-121.