Ensuring Credibility of Online User Reviews

Bipasha Banerjee
Computer Science
Virginia Tech
Blacksburg, VA, USA
bipashabanerjee@vt.edu

Smridh Malhotra
Computer Engineering
Virginia Tech
Blacksburg, VA, USA
smridhm@vt.edu

Ramya Nandigam
Computer Science
Virginia Tech
Blacksburg, VA, USA
ramyan@vt.edu

1. INTRODUCTION

Online reviews about various products and services are invaluable resources for Internet users and provide a rich source for users to make judgements about these products. Unfortunately, various forums such as Amazon, Yelp, TripAdvisor and others are being increasingly affected by manipulative and deceptive reviews. The reviews can be classified as fake (in order to promote/demote something), biased (providing an inconsistent view of the item), or irrelevant.

Research efforts have been made to automatically detect non-credible reviews.[1] Some methods that currently exist in the industry are considered to be fairly crude; for example, reviews that exploit data about various user activities (such as number of reviews posted and users with ratings of reviews with high standard deviation compared to the mean) [2] are marked as suspicious. The methods that currently exist place more emphasis on, and give more credibility to, long-term users. Moreover, these filters also employ several aggregated metadata, and are thus hardly viable for new items that initially have very few reviews.

State of the art methods include research on this topic that has cast the problem of review credibility into a binary classification task: a review is either credible or deceptive. [3] To this end, supervised and semi-supervised [4, 5] methods have been developed that largely rely on features about users and their activities as well as statistics about item ratings. But due to the subjective nature of trust, we don't want to classify the users; instead we want to provide global credibility values.

In this context, attempting to identify non-credible reviewers providing inauthentic reviews is superior to attempting to identify inauthentic reviews themselves. This is because non-credible users can easily generate a virtually unlimited number of inauthentic reviews if they are not banned from participating in the network. Therefore, we aim to build a web of trust, in which each user *i* will have been assigned a global trust value, which reflects the experience of all users in the network with user *i.* In our approach all users contribute distributivity to compute these values in a symmetric manner.

However, one problem we encountered while building such a *user-user* graph based on trust values is that e-commerce datasets provide reviews, the users who posted them, and the number of people who found them useful, but it does not state who found them useful. Thus, we observe people purchasing products or adopting particular behaviors without explicitly knowing who was the influencer that caused the adoption or the purchase. In order to address this, we plan to statistically approximate the user-user graph, based on timestamps and usefulness scores given by the dataset.

2. PROBLEM DEFINITION

In this study, we build a user-user web of trust, in which each user *i* will be assigned a global trust value, which reflects the experience of all users in the network with user *i.* We statistically approximate the user-user graph, based on review timestamps and usefulness scores given by the dataset.

3. RELATED WORK

We first started looking at the available datasets provided by our instructor and gathered an idea about what kind of data we would like to use for our project. We found the SNAP dataset quite interesting and wanted to delve into solving problems associated with the same. There has been some work done using the SNAP dataset and we wanted to look into what kind of research questions they solved in the dataset.

The paper by Richardson et al. [6] proposes that as the trust of a user is a subjective issue, instead of assigning credibility values to users by classify ing on the basis of information about items/users, the focus should instead be on building a web of trust for users, in which each user maintains a set of trusts, which may vary from user to user. The paper suggests two methodologies: Path Algebra Interpretation and Probabilistic Interpretation, to combine values of trusts using combination and aggregation methods. The system is based on two terms: belief and trust. Personal belief is defined as the belief of a person in a particular statement ranging from 0 to 1. Personal trust is defined as the trust of a user on any other user, and also ranges between 0 and 1. Using operations of merging (aggregation and concatenation) in the web of trust, merged beliefs and targeted trusts are computed. This results in providing a set of trusted sources for each user. The authors of this article carry out their experiments using BibServ, which is a bibliography service. A good entry for a user is one which has all the fields filled out and matches the research of the user, and thus is supposed to have high belief. Thus, each entry is a statement, based on which the personal belief matrix is constructed. personal trust matrix is built on user ratings. These merged beliefs and trusts yield a trust network containing users and their set of trusted users.

The paper by Kamvar et al. [7] talks about EigenTrust and the reputation in the P2P networks. They described an algorithm which would help decrease the download of inauthentic files in a P2P network. They introduced the concept of global trust that reflects the experience of all other nodes in the network about a particular node *i*. The global trust value is calculated based on the local trust value assigned to node *i* weighted by the global trust value of the assigning nodes. The global trust is calculated based on the left principal eigenvector matrix of normalized local trust values and thus the entire system is taken into consideration. EigenTrust serves as something of a benchmark in trust management, and many more recent papers build upon the strides made by the authors of the EigenTrust paper. Therefore, we are basing our calculations on it and plan to build on it.

The paper by Gomez-Rodriguez et al. [8] talks about diffusion and propagation of various cascades in a network. While it is easy to see when a node in the graph got infected by the cascade, it is difficult to assert who infected whom, as the underlying graph over which the diffusion took place is actually unobserved. The paper tackles this by tracing the paths of diffusion and inferring the graph over which the diffusion of cascades took place. Given the times of nodes getting infected, they develop a priori-model to trace how diffusion occurs from node to node; in the process it is seen that diffusion occurs in form of a tree, and thus the problem is a special case of maximum spanning tree. Therefore, naively searching for such a tree in the graph would require exponential time. Moreover, finding an optimal solution to the problem is NP-Hard. Thus, the paper lays out an approximation algorithm to find a near-optimal solution.

4. PROPOSED METHOD

Part A: Intuition/Proposed Work

We tackled the problem laid out by [6] and [7] in two steps. As the Amazon Dataset contains the timestamp when a review is posted which can be seen as when a node is infected with a cascade (influence of a review by another user), but it does not give the information about who found whose review useful, essentially the user to user relation is missing. Therefore, the first step is to construct the user-user influence graph, based on the approach in [6]. The second step is to assign global trust values based on the approach laid out by [7].

In step one, we essentially tried to estimate the structure of the user-user network by building a priori-model on the timestamp and attribute (usefulness). If a user *u* posted a review at time *t1,* it is likely that another user *v*, might base his/her decision to buy that product based on the review given by *u*, which is therefore dependent on the usefulness of the review by *u*, and the time difference between *u's* review and *v's* review. Based on these we developed a priori function to lay out given that the review by *u* is posted the probability that *v*'s decision and therefore *v*'s review was influenced by *u*'s review. Defining on these edges we see that diffusion of influences is in the form of a tree. Therefore, the problem is a special version of maximum spanning tree which is a maximum weight directed spanning tree. Since this is an NP-hard problem to solve, we found near optimum solution using methods laid out in [8].

In step two, we assigned a global trust value to each node based on the local values assigned to that node by its neighboring nodes which is weighted by the global values of the assigning nodes. The trust value in our constructed graph is the edge value from one user to another pertaining to a particular product. We accumulated these edges to find a single value between users *u* and *v*. Then local trust values are obtained by normalization procedures to avoid assigning false higher trust values to and by malicious nodes. To aggregate the normalized local trust, we looked at node *i'*s neighbors in the graph and took an opinion about their neighbors. This opinion is also weighted upon the trust each user has on its own neighbor. Thus, in this manner we can obtain global trust values for each user. We have done this in a distributed manner for faster computation.

Part B: Challenges and Assumptions

The optimal solution provided in [8] rests on the assumption that the underlying graph of diffusion will be Directed Acyclic Graph (DAG) in that if node *u* is infected before node *v* i.e., $t_u < t_v$ then only edge will have a weight; otherwise it will be 0. For example, when talking about blog influences user *u* writes a blog which user *v* cites, there is no possibility that user *u* will cite that in the future (the blogs are considered static).

But in our problem, in the cumulative graph constructed considering reviews for all products, it may be possible, that user *u* writes a review which influences user *v* for a product *x*, such that $t_u < t_v$ while user *v* writes a review which influences user *u* for a product *y* such that $t_v < t_u$. In this case we obtain a cycle. Therefore, it is not possible to solve for this graph, as it will be an NP-hard problem.

The solution we have for this is that we estimate a graph for each product separately. Therefore, the suboptimal graph for each product can be obtained using the laid-out approach as it will be a directed acyclic graph (DAG). Later on, we take a union of these graphs, before proceeding with Step 2. As for Step 2, we anyway accumulate all the edges between given two nodes.

Part C: Formal Description of Algorithm

Structure learning of the User-User influence graph

We define a cascade as a review posted for a particular product. Hence c = (t,P_s) where t is the Unix time at which the review is posted by a particular user and P_s is the usefulness score defined as the number of people who found the review useful divided by total number of votes for a single propagation tree. Considering a single edge (u,v) given the times (u,t_u,s_u) and (v,t_v,s_v) with $t_u < t_v$ The probability that *u* spreads the cascade to *v* i.e., $P_c(u,v)$. We assume $t_u > t_v$ and we set $P_c(u,v) > 0$. We state that this probability depends on the time difference between the two defined by $\Delta = t_v - t_u$ and the usefulness score. P(Δ) can be defined using either exponential or power-law model, each with parameter α.

$$P(\Delta) \propto e^{-\frac{\Delta}{\alpha}} \; and \; P(\Delta) \propto 1/\Delta^{\alpha}$$
$$P_c(u,v) = P(\Delta).P_s$$

We define probability of observing cascade c in a given tree structure t as

$$P(T) = \prod_{(i,j)\in T} P_c(i,j)$$

The probability that cascade c occurs in a graph G is given by

$$P(G) = \sum_{T\in T(G)} P(T)P(T|G)$$

We now define probability of a set of cascades C occurring in G

$$P(G) = \prod_{c\in C} P(c|G)$$

Therefore, the optimization problem we want to solve is

$$G' = arg \; arg\max_{|G|\le k} P\,(C|G)$$

Instead of considering every possible spanning tree, we arrive at an efficient solution. We take the log-likelihood for cascade *c* as $F_C(G)$, and arrive at the following for a set of cascades:

$$F_C(G) = \sum_{c\in C} F_c(G)$$

We define $F_c(G)$ as

$$F_c(G) = \sum_{(i,j)\in T} w_c(i,j)$$

Where $w_c(i,j) = logP_c(i,j) - log\;log\;\varepsilon$ is a non-negative weight which is the improvement in log-likelihood of (*i, j*) under the most likely spanning tree *T* in *G*. ε is defined as the probability that a particular user referred to an external source instead of some other user's review, and is pre-set to a small value. Therefore,

$$G' = arg \; arg\max_{|G|\le k} F_C(G)$$

Computation of global trust values

So far, we have found out the graph for each product. We will take a union of these graphs. Now each of the edges between users *u* and *v* correspond to one particular transaction between *u* and *v* for a certain product. Therefore,

$$s_{uv} = \sum tr_{uv}$$

Then we normalize these edges as

$$c_{u,v} = \frac{max\,(s_{uv},0)}{\sum_v \; max\,(s_{uv},0)}$$

Now to aggregate the normalized local trust values, for each user *i* we ask its neighbors about their opinion of other users. Therefore, logical way to write this would be

$$t_{uw} = \sum_v c_{uv}c_{vw}$$

Where t_{uw} is the represents the trust that node *u* places in node *w* based on asking his neighbors. We can write it in the matrix form as $\overrightarrow{t_u} = C^T\overrightarrow{c_u}$. To find global trust values, each user will have to take into consideration all other *n* users. Therefore, the equation will be:

$$t = (C^T)^n c_u$$

In order to improve the speed of the computation, we will need to do the above calculation in a distributed manner.

Part D: Implementation Methodology

Implementation of NETINF Algorithm

We found the open-source implementation of the NETINF algorithm [8]. On the basis of this code, we built our solution to meet the specifications of our input and problem statement. Based on the user reviews given by each user for one particular review, we connected these users in the graph. From this basic bare bone graph, we tracked the influence of a review by one user to another user and retained the most probabilistically strong relations. The probabilistic decisions were made on the basis of random walks on the graph, along with time difference between the user reviews and the local usefulness of these reviews we tested the probabilistic model for time difference as both exponential- and power-law based. With these parameters we implemented a Network Influence Model for this dataset in Java. Our full implementation can be found in our

Github repository: https://github.com/Bipasha-banerjee/ReviewTrustNet.

<u>Implementation of EigenTrust Algorithm</u>

After obtaining the user-user network, we computed the global trust values for each user in a fashion where each user asks its neighbors about their opinions of neighbor reviewers. We begin doing this by rating the transactions among each user marked in the graph, based on the usefulness of the source they are connected to. If node j follows a review of node i, then this transaction is marked as +1 or -1 as per the usefulness score.

Based on these transaction scores, we obtain the global trust scores which are 2nd-norm eigenvalues of the transaction matrix C described above, based on the Eigentrust Algorithm described in [7]. Our full implementation can be found in our Github repository: https://github.com/Bipasha-banerjee/ReviewTrustNet.

5. RESULTS AND DISCUSSION

Through our research, we aim to answer the following questions:

- What user-user relationship exists in online recommendation systems?
- What is the global trust of a specific user?
- Can we determine the trustworthiness of a user's review based on the inferred user-user network?

Part A: Dataset Used

We have decided to use the SNAP dataset on Amazon Musical Instruments reviews. It is essentially a web data of online reviews. It consists of 10,261 reviews and 1,429 users. The data includes various features of Amazon reviews, which include product ID, product title, product price, reviewer ID, reviewer name, fraction of users who found the review useful (usefulness score), rating of the product, time of the review, review summary, and the review itself. All of these will be useful to us, but the features which will be most useful to us when inferring the network are the usefulness score and the time of the review. The time at which the review was posted is given in Unix time. This information can be used to approximate the edges in our inferred user-user network. The usefulness score is given in the following format: [x, y], where x is the number of users who found the review useful and y is the total number of votes on the usefulness of the review. This is crucial in providing a priori attribute score, which can be used to support the timestamp-based approximations to find the user-user influence network as proposed. A sample illustration of the user-product bipartite graph which can be constructed is shown in Fig. 1.

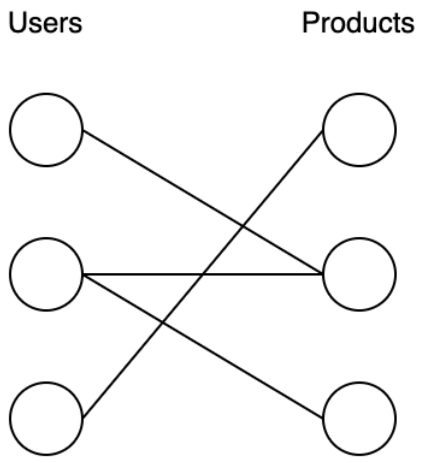

Figure 1: Sample user-product bipartite graph

Part B: Evaluation

Based on the NetInf approach described above, we obtained the graph in Fig. 2 which is comprised of 1325 nodes and 3352 estimated edges between them, comprising various transactions over different products. However, as per the dataset not all reviews had a usefulness score available. Thus, with the least value assumed to be 0.01, we constructed this graph. However, even with this value being lesser than the available scores, when multiplied with the time diffusion, the probability proved to be comparable when the time difference of these non-useful reviews from other reviews is reasonably less.

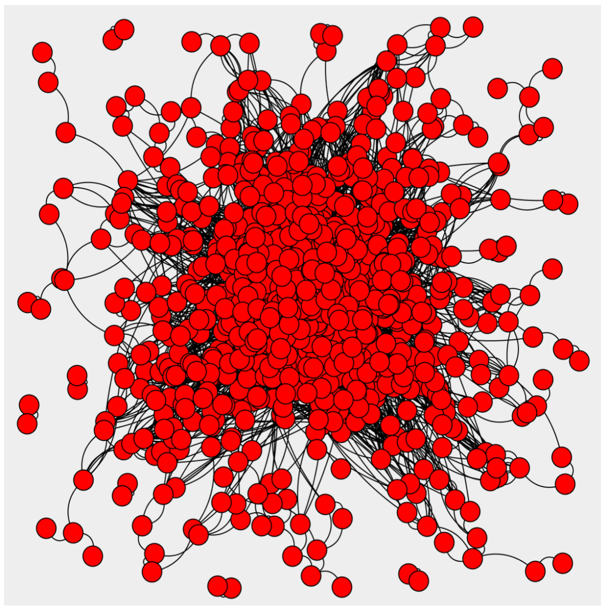

Figure 2: Inferred larger user-user network

Therefore we, in further iterations, removed these non-useful reviews to get a better sense of the cascades. In this graph we obtained 675 nodes and 690 edges indicating that the data was heavily biased with non-useful reviews. In this dataset we obtained only the edges with high product of time difference probability and usefulness scores ranging from 0.33 to 1. This graph is laid out in Fig 3.

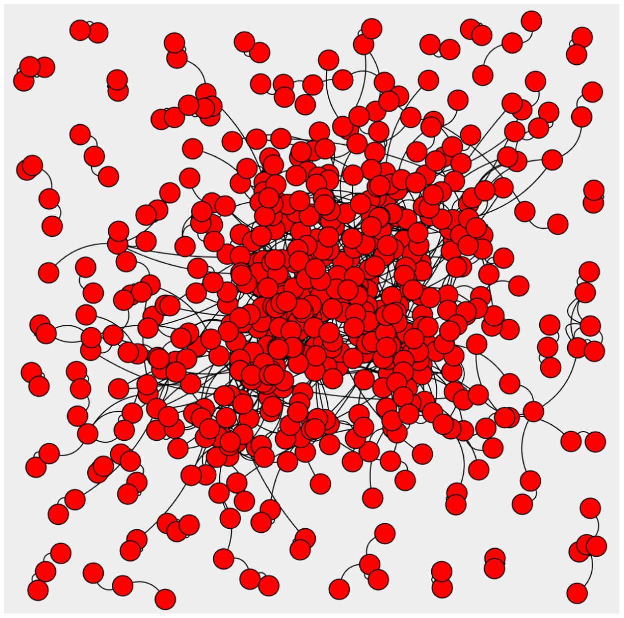
Figure 3: Inferred smaller user-user network

We implemented the model with both exponential- and power-law time diffusion models. Both of the graphs correctly represent the model, but however the power law scales better with bigger values, as in our case the time values we used were Unix Time values which are the number of seconds since Jan 1, 1970. With the exponential model, the probabilistic values with a time difference of the order of 10^6 seconds, tends to infinity. Therefore, we had to scale up these values with appropriate choice of alpha as well as changing units. However, the power law model scaled well with bigger values and required smaller alpha values and time units.

On both of these networks we performed the EigenTrust calculations to obtain the global trust values. With the graph including non-useful reviews, we obtained trust values truly dependent only on pre-trusted users defined by us. This was: all the transactions from a user with a non-useful score will have a 0.01 value. Upon thresholding these values are marked as negative, and thus as per the algorithm are zeroed out on totaling. So they are based only on the pre-trusted score parameter. With these trust scores it was hard to decipher a user as trustworthy or not.

In the reduced graph, we were able to find more credible values, as only the nodes with all transactions rated negative were zeroed out. And only the trust scores of these edges were later reduced to depend on the pre-trusted users in the network, thus marking these respective transactions as non-credible.

We then compared the global trust scores obtained to the usefulness scores available in order to classify a good review from a bad one, where classification on the usefulness scores takes into account only the scores and ratings of one particular review, while the global trust values account for experience of all the users in the network. We used a separate test set with 400 reviews with valid usefulness scores. The results obtained by local usefulness scores marked 335 reviews as credible and 65 reviews as non-credible, whereas the global trust-based classification marked 301 reviews as credible and 99 reviews as non-credible. We believe that since it takes in all the experiences of various users, the scores are a bit more realistic, and thus on keeping the same threshold we see the reduce in the credibility.

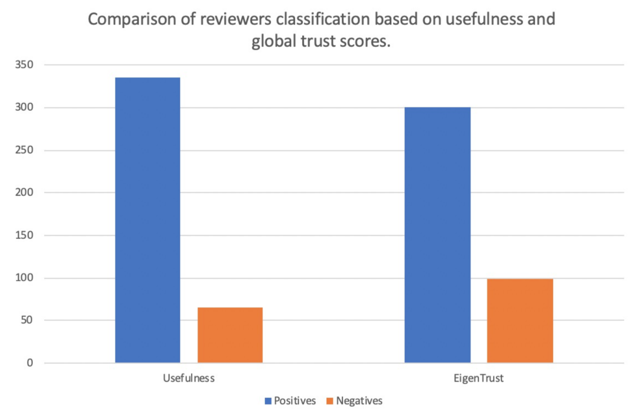

Figure 4: Classification of reviewers based on usefulness values and global trust scores

6. CONCLUSION AND FUTURE WORK

Overall, through the NetInf algorithm we were able to obtain a user to user relation by the probabilistic approach. The EigenTrust value assigned a global score to the user and thus we obtained the “trust” of a particular user.

There are some ideas that we would like to work on in the future. We would like to further implement our algorithm on datasets that would help us predict more accurately.

Additionally, it would be good to use neural network models to actually train and then test on the dataset. This would help us measure the accuracy of our model and can be implemented in real world applications.

7. REFERENCES

1. N. Jindal and B. Liu. Opinion Spam and Analysis. *Proceedings of the 2008 International Conference on Web Search and Data Mining*, 219-230, 2008.
2. A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance. What Yelp Fake Review Filter Might Be Doing? *Proceedings of International AAAI Conference on Web and Social Media*, 2013.
3. S. Mukherjee, S. Dutta, and G. Weikum. Credible Review Detection with Limited Information Using Consistency Features. *Machine Learning and Knowledge Discovery in Databases*, 195—213, 2016.
4. H. Li, Z. Chen, B. Liu, X. Wei, and J. Shao. Spotting Fake Reviews via Collective Positive-Unlabeled Learning. *Proceedings of the 2014 IEEE International Conference on Data Mining*, 899—904, 2014.
5. H. Li, Z. Chen, A. Mukherjee, B. Liu, and J. Shao. Analyzing and Detecting Opinion Spam on a Large-scale Dataset via Temporal and Spatial Patterns. *International AAAI Conference on Web and Social Media*, 2015.
6. M. Richardson, R. Agarwal, and P. Domingos, Trust Management and Semantic Web. *Proceedings of the Second International Conference on Semantic Web Conference*, 351—386, 2003.
7. S. Kamvar, M. Schlosser, and H. Garcia-Molina. The Eigentrust Algorithm for Reputation Management in P2P Networks. *Proceedings of the 12th International Conference on World Wide Web*, 2003.
8. M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring Networks of Diffusion and Influence. *ACM Trans. Knowl. Discov. Data*, 2012.