Analysis of Trust from Epinions.com

Reed Halberg
James Kerr

December 5, 2013

1 Introduction

For this project we analyzed data from Epinions (epinions.com) provided at the following url: http://snap.stanford.edu/data/soc-Epinions1.html

The website epinions.com is, simply, a review hub, where users may rate and comment on any product, as well as read the opinions about products posted by other users. Epinions.com is strictly a website for reviewing products; it sells nothing (though, it does link to other websites that do sell the items).

One feature of Epinions is *trust*. A user may decide that some other user posts particularly insightful reviews, or has a similar taste (etc.), and wants to stay connected for any future posts. This user may choose to trust the other user, establishing a link between them and tailoring his or her experience to satisfy all product review needs.

This trust can be modelled in a directed graph, each vertex representing a user and each edge representing the one way action of trust. We decided to examine the data provided to gain some insight into what features are similar among very trusted users. The results of such an analysis may be very interesting to users wanting to be more popular on the site or the site's owners themselves.

Specifically, we looked at a few things: 1) if more trustful users tended to have more reciprocal trusts (given trust that was returned) on average; 2) if the most trusted users tended to receive more reciprocal trust on average; 3) if the most trustful nodes were very trusted by others on average.

A reciprocal trust is a trust that occurs in both directions; a user trusts another user who trusts the first user back. We considered reciprocity to be the ratio of number of reciprocal trusts received to number of trusts given by a user, or the percentage of trusts given that were returned.

See our implementation details to see how we examined the data for the above information.

2 Implementation

Upon running, the program will load data from the data file specified as a program argument. Then the user is prompted with a text-based user interface that allows the

user to ask specific questions about the graph or to generate data files that could then be further analysed. Since all of the questions we were asking centered around similar data (primarily, how many trusts were going into a user or going out from a user), we decided that it would be easiest to collect the data that would be useful to us in a comma separated value (.csv) file, then utilize analysis tools built into Microsoft Excel to extract meaning. We concluded that due to the relatively small size of our dataset (approximately 75,000 Users), analyzing the data in Excel was reasonable.

Our program uses a map data structure from the standard library to hold the information needed. The map works as an adjacency list, each ID number from the inputted data corresponding to an ID in the map.

Each element in the map was a custom object called a **User**, which held the fields: ID number; list of other users that **User** trusts; number of other users that trust **User**; number of reciprocal trusts (other users that **User** trusts which also trust the original **User** back). As the data file was read, the fields were managed so that no additional processing was needed to construct our representation of the graph; however, the reciprocal trusts were calculated during the creation of the .csv file.

At this point, all the desired data was held in the map, each User object containing the data we wanted to export to the .csv file. So, we created a method that would facilitate printing the fields in the User class to a file, and created another method that printed all the Users together. This created the .csv file. The Excel analysis of this file will be discussed in the Results section below.

To find if two users were connected and the shortest path between them, we used Dijkstra's algorithm, keeping track of the costs and parents between nodes.

3 Results

Before discussing the results of our primary analysis, we will mention some preliminary information about the dataset. First, there were in all, 75,879 users (vertices) and 508,836 trusts (edges). The average number of trusts from a user and to a user were both 6.7059 (since the total number of trusts given equals the total number of trusts received, the averages must be equal). The greatest number of trusts from a user was 1801. The greatest number of trusts received by a user was 3035.

To analyze reciprocity, we used the columns in the .csv file as parameters to calculate the linear regression between different variables. Our results are as follows:

1) $R = 0.00134T + 0.2587$

where T is the number of trusts given by a user and R is the reciprocity. This equation had a correlation of 0.104 to the data.

2) $R = 0.0016I + 0.2570$

where I is the number of trusts received by a user and R is the reciprocity. This equation had a correlation of 0.159 to the data.

3) $I = 0.7238T + 2.1367$

where T is the number of trusts given by a user and I is the number of trusts received. This equation had a correlation of 0.551 to the data.

4 Conclusion

Given the low correction for our first equation (0.104), which compares trusts from a user to reciprocity, we can conclude that a user *will not* generate higher-than-average feedback for trusting a user, for no reason other than having many outward trusts already.

Similarly, the second equation, with a nearly equal correlation (0.159), we can see that as a user receives more trust, there is no indication that the user is also receiving a greater proportion of reciprocal trusts.

Even though a strong correlation would have provided users an option to generate more trust (if they wanted more trust received, they would only have to trust many others, who would then reciprocate), it does provide insight into the social dynamic of the sight. Users tend not to respond to the 'status' of a user who decides to trust them, meaning, if a user, A, is very-trusted, and trusts another user, B, it is not likely that B will not respond simply because many other users trust A. Instead, perhaps, B's reciprocity will only occur if B decides A is worth trusting on some other terms, like quality of ratings or taste.

Finally, with the third equation, it is apparent that there does, in fact, exist a weak connection between the number of trusts given by a user, and the number of trusts received by that user (correlation of 0.551), and that the relation is proportional. The meaning of this is not entirely clear without further analysis, or more data about the nodes. It is possible, for instance, that users that trust more, are more visible on the website due to its internal mechanics, and thus get more attention directed to them and more trust. It is possible that a high number of outward trusts coincides with users that are more active in general, and also rate more products, again making it easier for other users to stumble across their profiles.

We do know, with a reasonable correlation, that as the number of trusts given by a user increases, so do the number of trusts received.