POLARITYTRUST: MEASURING TRUST AND REPUTATION IN SOCIAL NETWORKS

F. Javier Ortega, José A. Troyano, Fermín L. Cruz and Fernando Enríquez de Salamanca

Department of Computer Languages and Systems, University of Seville, Spain
javierortega@us.es
troyano@us.es
fcruz@us.es
fenros@us.es

ABSTRACT

In this work we tackle the problem of determining the trustworthiness of the users in a social network. Our approach introduces the novelty of taking into account the negative opinions in a social network to obtain the ranking of trust according to the opinions of all the users in the network. We briefly discuss some common attacks that malicious users can perform against a system in order to gain good reputation in the network. The experiments are performed with synthetic graphs, randomly generated to model real social networks according to some common features, and to simulate the attacks previously mentioned. The results show that our approach can deal with these threats, demoting malicious users and minimizing their effects in the final ranking of trust.

KEYWORDS

Trust and Reputation, Graph algorithms, Social Networks

1. INTRODUCTION

Social networks have experimented a great expansion in the past few years, covering a wide variety of themes and functionalities, and allowing their users to share many kinds of contents and to establish different types of relationships between them. One point in common for the majority of the social networks is the necessity of qualifying their own contents in order to provide a better service and to improve the user experience. In social news sites and other content-sharing networks is very useful to take advantage of the user opinions in order to give more relevance to some contents over others. In on-line marketplaces is crucial to distinguish untrustworthy sellers or buyers, so these systems usually allows their users to evaluate their transactions. Most of the systems provide the users with the ability of giving their opinions about other users, or the contents generated by them. The problem comes out when a user or a group of users take advantage of the voting system in order to gain any kind of benefits. For example, in an on-line marketplace, a dishonest seller would want to gain high reputation in order to increase his sales. These actions can provoke negative consequences in the services provided by these sites, disturbing the normal behaviour of the social networks. *Trust and Reputation Systems* (from now on *TRS*'s) are intended to deal with this problem, avoiding the effects that users with dishonest behaviours can cause in a social network.

Since this problem is very important for most of the social networks, it has been treated in many works from different points of view. Many social networks implement their own TRS, intended to deal specifically with the dishonest users in their systems. In this work, we propose a novel approach to this task, introducing a general method that takes advantage of both positive and negative opinions of the users in a social network, in order to build a ranking of users according to their trustworthiness. Our approach is intended to demote in the ranking the users who present a dishonest behaviour in the system. As far as we now, there are not many works on trust and reputation that process a network with negative opinions between their users.

The rest of the work is organized as follows. In Section 2 we briefly talk about other works on trust and reputation analysis. Some common attacks against TRS's are discussed in Section 3. Section 4 introduces our approach and one variation proposed to deal with the different attack models. The design and the results of the experiments performed to evaluate our proposal are shown in Section 5. Finally, in Section 6 we point out the conclusions and future work.

2. RELATED WORK

Recently, many works about TRS's have been carried out, studying the challenges that these systems must face in order to provide the users of social networks with reliable information about the trustworthiness of the rest of the users in the system. Common problems in the implementation of TRS in a social network are discussed in [2] and [3]. They point out the existence of a bias in the majority of the ratings toward giving positive scores. They also talk about the absence of incentives that users usually have for providing ratings in the system.

In [4] and [5], a set of common security vulnerabilities for TRS's are identified: the initial window problem (or cold start) occurs in TRS's that relies only on the user direct experiences, so new users does not have any information about the trustworthiness of the users in the system, and vice versa; the re-entry problem, which points out the impossibility of establishing the identity of a user, allowing one user to create several accounts in the system to favour one to another; finally, the exit problem consists in the negative behaviour that can present a user who is planning to leave the social network, and who has no further need for his good reputation. Most of them are difficult to avoid, so it is a good approach to try to minimize their negative effects.

Some proposals for methods intended to deal with the computation of trust and reputation in networks are presented in [1], [6], [7], [8], [9] and [10]. For example, in [1], Guha et al. propose a framework in order to propagate trust and distrust in a network, with ratings between some nodes. The goal is to compute the belief matrix, F, taking as input the initial beliefs matrix, B. B is based on the trust and distrust relations between the nodes. F is calculated by performing multiple steps of atomic propagations. *Atomic propagations* extend a conclusion by a sequence of steps in the graph of trusts.

EigenTrust [6] is an algorithm that calculates global trust values for users in P2P networks. In the EigenTrust algorithm, each peer calculates the local trust value for all peers that have provided it with authentic or fake downloads. The global score is obtained by aggregating the normalized local trust values with respect to a peer. In [7] they present some extensions to the EigenTrust algorithm.

3. THREAT MODELS IN TRS'S

In this section we study some common tactics that can be adopted by users who want to gain some kind of benefits from the TRS. Several threat models are presented in [6] and [7]. They take the example of a P2P network for file sharing in order to explain the methods used by malicious users to achieve their goal. In many senses, a social network can be viewed as a P2P network, in terms of a decentralized network where users can share different resources (texts, videos, images, etc). In other words, social networks can be attacked in a similar way as P2P networks. According to these works, the main threat models to interfere in the overall ranking of trust can be described as follows:

- **Threat Model A:** *Individual Malicious Peers*. Malicious users always present a bad behaviour, so they receive negative links from good users. In fact, this model represents the absence of attacks against the network, because the behaviour of each type of user is just as expected, so the ranking of trustworthiness is not affected.

- **Threat Model B:** *Malicious Collectives*. Based on previous model, adding the possibility of bad users to assign positive trust values to other malicious users. In this way, the ranking of malicious users can be increased due to the amount of positive in-links received.

- **Threat Model C:** *Camouflage behind good behaviour*. In this attack, malicious peers can cheat some good users to vote positively for them. The effect in the network is that some bad users can received sporadically a positive vote from a good user.

- **Threat Model D:** *Malicious spies*. There are two types of malicious users: some of them acts as in threat model A or B; and the others, called *spies*, who make good users to vote positively for them, but assign positive trust values only to bad nodes.

- **Threat Model E:** *Camouflage behind judgments*. In this model, malicious peers assign negative votes to good peers. This strategy can cause the decrease of trust of good peers and, as a consequence, the promotion of the malicious peers in the ranking of trust.

4. POLARITYTRUST

In this section we introduce our approach to the problem of determining the ranking of users in a social network with positive and negative opinions between them. We also present an extension of the basic model in order to deal with some common vulnerabilities in TRS's.

4.1. Definition

PolarityTrust is a graph-based ranking algorithm inspired by PageRank [11]. It adapts PageRank in order to handle graphs with positive and negative edges. PolarityTrust defines two different ranking values for each node in the graph, *Positive PageRank* (PR^+) and *Negative PageRank* (PR^-). Formally, let $G = (V, E)$ be a directed weighted graph with a set of vertices V and a set of directed edges E. Given two nodes, v_i *and* $v_j \in V$, we define p_{ij} as a real valued attribute that represents the weight of the edge from v_i to v_j, with $p_{ij} \neq 0$. For a given vertex v_i, let $In(v_i)$ be the set of vertices that point to it. And let $Out(v_i)$ be the set of vertices that v_i points to. The scores can be obtained as it is shown in Equations (1) and (2).

$$\begin{aligned} PR^+(v_i) \;=\; & (1-d)e_i^+ + d(\\ & + \sum_{j \in In^+(v_i)} \frac{p_{ij}}{\sum_{k \in Out(v_j)} |p_{jk}|} \; PR^+(v_j) + \\ & + \sum_{j \in In^-(v_i)} \frac{-p_{ij}}{\sum_{k \in Out(v_j)} |p_{jk}|} \; PR^-(v_j)) \end{aligned} \tag{1}$$

$$
\begin{aligned}
PR^-(v_i) &= (1-d)e_i^- + d(\\
&+ \sum_{j \in In^+(v_i)} \frac{p_{ij}}{\sum_{k \in Out(v_j)} |p_{jk}|} PR^-(v_j) + \\
&+ \sum_{j \in In^-(v_i)} \frac{-p_{ij}}{\sum_{k \in Out(v_j)} |p_{jk}|} PR^+(v_j))
\end{aligned} \tag{2}
$$

where e^+ and e^- are the personalization vectors. They are intended to cause a bias in the algorithm. In PolarityTrust, they contain the nodes which opinions should be more relevant *a priori*, such as the social network administrators, or other kind of authority of the website.

The ranking of nodes is built according to the difference between PR^+ and PR^- for each user.

4.2. Non-Negative propagation (NN)

Malicious users have many ways to take advantage of the weaknesses of the ranking algorithms. In order to avoid the influence of bad nodes in our system, we integrate in the PolarityTrust algorithm the ability of deciding whether the opinion of the users must be taken into account or not. Thus we can minimize the influence of bad users in the ranking by allowing only specific opinions to be propagated over the network. This feature is very useful to deal with some threat models, because the ranking can be protected from the opinions of malicious users.

The intuition says that the opinions from good user must be always taken into account, whereas some opinions from bad users must be avoided because they can cause negative effects in the final ranking. In the case of our TRS, the positive opinions from bad users are very useful to determine the badness of other users, while negative opinions can cause the demotion of good users in the ranking. So the proposal consists in avoiding the negative opinions from bad users, in order to improve the ranking of users.

The problem of distinguishing whether a user is good or bad can be easily addressed using the node scores, PR^+ and PR^-. Our intuition says that a node can be considered a good user if its positive score is higher than the negative one. In this way, if $PR^+(i) < PR^-(i)$, then the positive votes from node *i* must not be propagated because it is a malicious user.

5. EXPERIMENTS

The experiments in this work are designed to show the reliability of our proposal with a set of randomly generated graphs, comparing the results of the different approaches to a baseline. Each graph has been generated including a specific threat model, or a combination of them, in order to test the vulnerability of our proposal against them.

5.1. Datasets

Due to the lack of publicly accessible datasets of social networks with positive and negative opinions between the users, we need a method to randomly generate graphs with a topology similar to real-world networks. One of the most cited studies on large real-world networks is presented in [12]. This work explains a common property of these networks: *Preferential Attachment*. It consists in the fact that the new users of a network attach preferentially to nodes that are already well connected. The work also introduces the Barabasi model, intended to generate graphs with the *Preferential Attachment* property. We use this method to generate the datasets for our experiments.

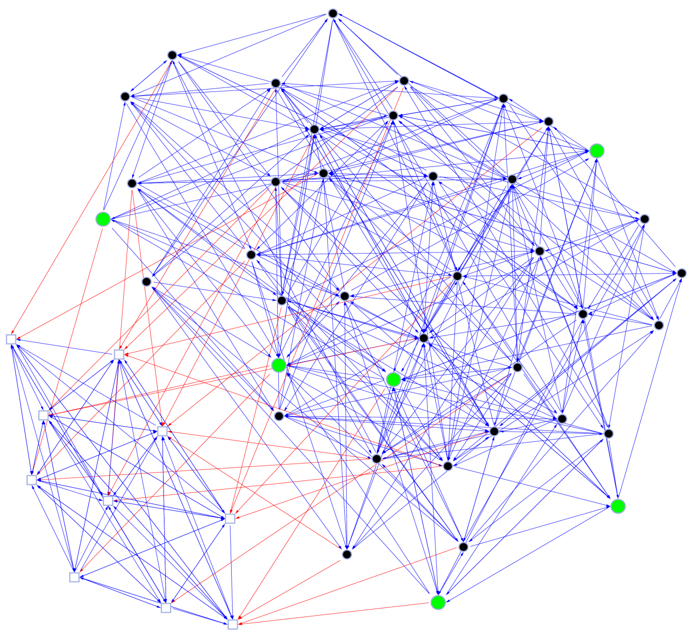

Figure 1: Graph generated by the Barabasi model. Squares represent bad users, and circles are the good users. Big circles represent seed nodes.

The method begins with a network of at least two nodes, and the degree of each node should be at least 1, in order to generate a connected network. New nodes are added to the network one at a time. Each new node is connected to a number of existing nodes with a probability:

$$p_i = \frac{k_i}{\sum k_j}$$

where k_i is the degree of node i. In this way, the degree distribution is a power law, and the network shows the preferential attachment property.

Since we want to determine the performance of our proposal against different types of malicious behaviours, we need to generate a graph model for each attack. The Barabasi model is implemented in order to generate the nodes and edges modelling the good user behaviours. The nodes and edges of malicious users have been generated according to each attack. In figure 1 we show an example of a random graph generated with this method.

For the experiments we have generated a set of graphs with 10^4 nodes, of which 10^3 are malicious users. In order to perform the attack model D (malicious spies), 10 nodes have been taken as spies, and 990 as bad users. A set of 10 nodes have been taken as the positive seeds for our algorithm. Intuitively, these seeds represent a special group of users, which opinions are totally trustworthy. We can think them as the administrators of the social network.

5.2. Baselines

In this section we present the techniques taken as baselines in the experiments. EigenTrust algorithm [6] aims to reduce the number of inauthentic file downloads in a P2P network. It computes the local trust value for all peers voting to each user. The global trust value is obtained by aggregating the normalized local trust values with respect to a peer. Formally, given *C,* a matrix where c_{ij} represents the opinion of *i* about *j* (local trust value). The algorithm computes the global trust values as:

$$\bar{t}_i = C^T \cdot \bar{c}_i$$

where $\bar{c}_i$ is the vector of local trust values of *i* for each node in the network. Repeating this process, $\bar{t}_i$ will converge to a stable value, $\bar{t}$, that is the vector containing the EigenTrust values for each node. This vector is the left principal eigenvector of the matrix *C*.

Fans Minus Freaks is a simple heuristic that takes into account the difference between the number of positive and negative votes of each user. It is obtained as follows:

$$FmF(i) = |\,\mathrm{In}^+(i)| - |\,\mathrm{In}^-(i)|$$

where In^+ and In^- are the positive and negative votes to *i*, respectively.

5.3. Comparative study

The experiments are intended to show the performance of our approaches and the baselines against the attacks seen in Section 3. Since the aim of these techniques is to demote the bad users in the ranking and to promote the good ones, we use the *"Bad Users in Good Positions"* as the evaluation metric. In other words, we evaluate the performance of the techniques in terms of the number of bad users that appear in the positions of the ranking corresponding to good users. The perfect system would rank the N bad users in the last N positions of the ranking, obtaining a metric of 0. So, the lower the metric value, the better is the performance of the technique. Note that, in this case, this metric is equivalent to the "Good Users in Bad Positions".

Table 1. Results for each method against basic attacks. Dataset of 10^4 nodes, of which 10^3 are malicious users. In model D we have included 10 spies in the network.

Models	ET	FmF	PT	PT+NN
A	50	0	0	0
B	197	36	0	0
C	63	207	94	94
D	86	9	9	9
E	74	4	0	0

We show in Table 1 the results for each isolated attack. We can see that model B is very effective against EigenTrust algorithm. FmF achieves better results for models B, D and E, though it is weak against model C. Our approaches, PT and PT+NN, perform very well against all the attacks, except for model C. In that case, some bad users can be taken as good ones due to some amount of positive votes from other good users. These votes can be made by mistake, or caused by the behaviour of these bad users.

In Table 2 we show the results of each technique against an incremental combination of attacks. We can see that more complex methods achieve better results than the simplest one. EigenTrust gets now better results than FmF, due to the complexity of the attacks. Our proposals have a very good performance, showing the usefulness of taking into account the negative opinions of users in the computation of the ranking of trust.

Table 2. Results for each method against incremental attack combinations.

Models	ET	FmF	PT	PT+NN
A	50	0	0	0
B	197	36	0	0
B+C	155	873	27	27
B+C+D	169	871	26	26
B+C+D+E	183	849	38	36

A graphical comparison of the techniques is shown in Figure 2. A set of random graphs have been generated modelling the combination of all the attacks, with a number of bad users varying between 100 and 2000. In the chart is represented the proportion of errors per bad user. Our proposals, PT and PT+NN, show a very stable performance for all the attack intensities, outperforming EigenTrust and FmF heuristic.

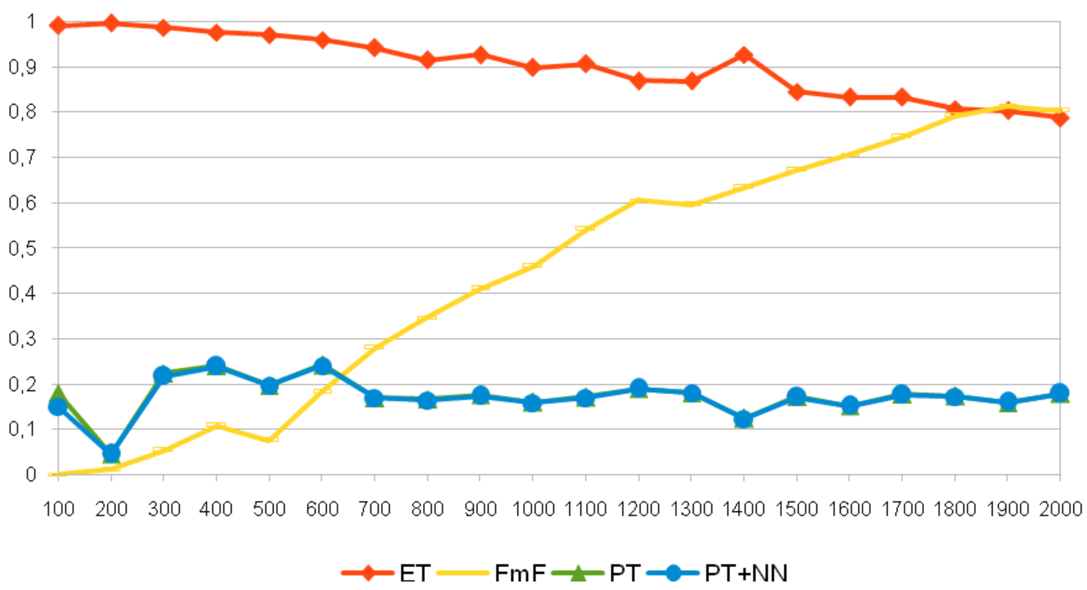

Figure 2: Results with attacks of 100 to 2000 bad nodes

6. Conclusions

In this work we have presented a Trust and Reputation System that builds a ranking with the users of a social network, regarding their trustworthiness. The novelty of our approach is that it takes advantage of the negative opinions of the users. The system has been tested with some common attacks that can be launched against a TRS. The results of the experiments show a good performance of our proposal demoting malicious users in the ranking of trust.

We plan to further our research by studying other types of attacks against TRS's, including the use of *playbook sequences*. This attack consists in a sequence of actions intended to gain high trustworthiness. There is an infinite set of possible playbook sequences, and they can be influenced by other user playbooks, making these attacks really hard to detect and to avoid. The intuition behind playbook attacks is that it cannot be assessed that a TRS is effective just because the potential attackers do not know how it works. On the other hand, we also plan to experiment with some test-beds for TRS's, such as [13], that provides us with a standard framework to test our approach and to compare its performance in a more realistic environment.

ACKNOWLEDGEMENTS

This work has been partially funded by the Spanish Ministry of Education and Science (*HUM2007-66607-C04-04*).

REFERENCES

[1] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In WWW ’04: Proceedings of the 13th international conference on World Wide Web, pages 403–412, New York, NY, USA, 2004. ACM.

[2] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. Decision Support Systems, 43(2):618–644, 2007.

[3] S. Marti and H. Garcia-molina. Taxonomy of trust : Categorizing p2p reputation systems. International Journal of Computer and Telecommunications Networking, 50(August 2005):472-484, 2006.

[4] R. Kerr and R. Cohen. Smart cheaters do prosper : Defeating trust and reputation systems the security of trses. In Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems, Budapest (Hungary), 2009.

[5] A. Jøsang and J. Golbeck. Challenges for robust trust and reputation systems. In 5th International Workshop on Security and Trust Management, Saint Malo, France, 2009. Elsevier.

[6] S. D. Kamvar, M. T. Schlosser, and H. Garcia-molina. The Eigentrust algorithm for reputation management in p2p networks. In Proceedings of the Twelfth International World Wide Web Conference, pages 640–651. ACM Press, 2003.

[7] D. Donato and M. P. Stefano Leonardi. Combining transitive trust and negative opinions for better reputation management in social networks. In Workshop on Social Network Mining and Analysis (SNA-KDD), 2008.

[8] J. A. Golbeck. Computing and applying trust in web-based social networks. PhD thesis, University of Maryland at College Park, College Park, MD, USA, 2005.

[9] C. de Kerchove and P. V. Dooren. The pagetrust algorithm: How to rank web pages when negative links are allowed? In Proceedings of the SIAM International Conference on Data Mining, pages 346–352, 2008.

[10] J. Kunegis, A. Lommatzsch, and C. Bauckhage. The slashdot zoo: Mining a social network with negative edges. In 18th International World Wide Web Conference, page 741, apr 2009.

[11] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web, 1999.

[12] A. L. Barabási and R. Albert. Emergence of scaling in random networks. Science, 286(5439). Pages 509-512, Oct. 1999.

[13] R. Kerr and R. Cohen. Treet: the trust and reputation experimentation and evaluation testbed. Electronic Commerce Research, 10(3-4):271–290, 2010.