A novel approach to propagating distrust

Christian Borgs[1], Jennifer Chayes[1], Adam Tauman Kalai[1], Azarakhsh Malekian[2], and Moshe Tennenholtz[3]

[1] Microsoft Research New England
[2] Northwestern University, EECS Department,
[3] Microsoft R and D center, Israel

Abstract. Trust propagation is a fundamental topic of study in the theory and practice of ranking and recommendation systems on networks. The Page Rank [9] algorithm ranks web pages by propagating trust throughout a network, and similar algorithms have been designed for recommendation systems. How might one analogously propagate distrust as well? This is a question of practical importance and mathematical intrigue (see, e.g., [2]). However, it has proven challenging to model distrust propagation in a manner which is both logically consistent and psychologically plausible. We propose a novel and simple extension of the Page Rank equations, and argue that it naturally captures most types of distrust that are expressed in such networks. We give an efficient algorithm for implementing the system and prove desirable properties of the system.

1 Introduction

Trust-based recommendation and ranking systems are becoming of greater significance and practicality with the increased availability of online reviews, ratings, hyperlinks, friendship links, and follower relationships. In such a recommendation system, a "trust network" is used to give an agent a personalized recommendation about an item in question, based on the opinions of her trusted friends, friends of friends, etc. There are many recommendation web sites dedicated to specific domains, such as hotels and travel, buyer-seller networks, and a number of other topics. In several of these sites, users may declare certain agents whose recommendations they trust or distrust, either by directly specifying who they dis/trust or indirectly by rating reviews or services rendered. We show how trust or distrust may be naturally propagated in a manner similar to Page Rank [9], extending the work of Andersen *et al.* [1], which was for trust alone. For simplicity, we focus on personalized recommendation systems, though our approach applies to ranking systems as well. We take, as a running example, a person that asks their trusted friends if they recommend a certain specialist. Some friends might have personal experience, while others would ask their friends, and so forth. The agent may then combine the feedback using majority vote or another aggregation scheme. Such personalized recommendations may better serve a person who does generally agree with popular opinion on a topic and may be less influenced by people of poor taste and spammers. Figure 1 illustrates two distrust networks, formally defined as follows. Among a set of nodes, N, there is a designated source $s \in N$ that seeks a positive, neutral, or negative recommendation. For $u, v \in N$, weight $w_{uv} \in [-1, 1]$ indicates the amount of trust (or distrust if negative) node u places in v, and we require $\sum_v |w_{uv}| \leq 1$ for each node u. Following Andersen *et al.*, for simplicity we consider two disjoint sets $V_+, V_- \subseteq N$,, of positive and negative voters, who are agents that have fixed positive or negative opinions about the item in question, respectively. In the case of a doctor, these may be the agents that have visited the doctor themselves. We assume that $w_{uv} = 0$ for any voter u, e.g., if they have first-hand experience they will not ask their trusted friends about the doctor. The simple random walk system [1] suggests that one consider a random walk starting at s and terminating at a voter. The recommendation is based upon whether it is more likely that the random walk terminates in a positive voter or a negative voter. In Figure 1, in case (b) clearly the recommendation should be negative, because the total trust in the positive voter is less than that in the negative voter, due to node z's distrust. However, in complex networks such as Figure 1, in case (c) it is not obvious whom to trust and distrust. It could be that w and z are trustworthy and y is not (hence y's distrust in w and z should be disregarded), or the opposite could be the case. It is not clear if there is any psychologically plausible and mathematically consistent recommendation system in general graphs.

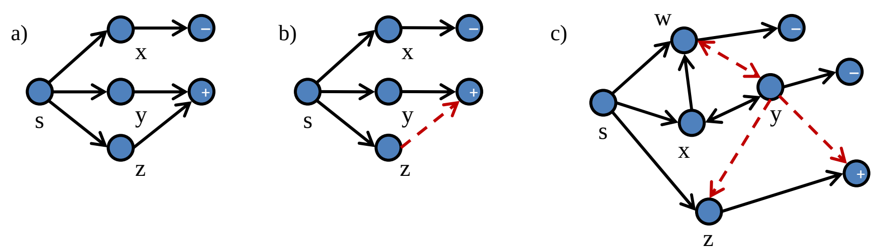

Fig. 1. Examples of (dis)trust networks. In each case, node s seeks a recommendation in $\{-, 0, +\}$. A solid/dashed edge from u to v indicates that u trusts/distrusts v. A $+/-$ indicates a voter has a fixed positive/negative opinion of the doctor (perhaps based upon first-hand experience). a) The recommendation is $+$. b) The recommendation is $-$. c) Typical networks may be cyclic and more complex.

We propose a simple solution, and justify it both through examples and the axiomatic approach, arguing that it satisfies a number of reasonable properties that one might like in a recommendation system. Moreover, it is the only system satisfying these properties. Before delving into the details, we point out that there are several types of people one might want to distrust in such a network:

1. **Bad guys**. Self-serving people, such as spammers, will recommend things based upon their own personal agendas. Distrust links can decrease their influence on such a system. One has to be careful, however, not to simply negate whatever they suggest, since they may be strategic and say anything to achieve their goal.
2. **Careless big mouths**. Say, in your social circle, there is a vocal and opinionated person who you have discovered is careless and unreliable. Even without a trust link from you to them, that person's opinions may be highly weighted in a recommendation system if several of your trusted friends trust that person. Hence, it may be desirable for you to explicitly indicate distrust in that person, while maintaining trust links to other friends so that you can take advantage of their opinions and those of their other friends.
3. **Polar opposites**. On some two-sided topics (e.g., Democrats vs. Republicans), a graph may be largely bipartite and distrust links may indicative that one should believe the opposite of whatever the distrusted person says.

We model distrusted people as behaving in an arbitrary fashion, and seek to minimize their influence on a system. This includes the possibility that they may be self-serving adversaries (*bad guy*), but it is also a reasonable approach for people whose opinions do not generally agree (*careless*). Our approach fails to take advantage of the opinions of distrusted people (*polar*), in two-sided domains. However, many real-world topics in which people seek recommendations are multi-faceted.

We believe that it is important for trust and distrust to be propagated naturally in such a system. For example, in case of *careless* above, the distrust you indicate will not only affect the system's recommendations to you but also to people that trust you. As we argue below, propagating distrust is a subtle issue. In this paper, we present a novel and simple solution which we feel is the natural extension of the random-walk system of Andersen *et al.* [1] (similar to Page Rank [9] and other recommendation system) to distrust.

The focus of the present paper is distrust propagation in domains where there is typically little overlap in personal experiences: it is unlikely that any two people have seen a number of doctors in common or been to a number of common hotels. Ideally, one might combine collaborative filtering with trust and distrust propagation, but we leave that to future work.

1.1 The propagation of trust and distrust

Figure 2 gives simple examples showing how trust and distrust may be used in recommendation systems. These examples illustrate why it is important to keep track not only of who is distrusted, but also who it is that distrusts the person. Roughly speaking, by trust propagation, we mean that if u trusts v and v trusts

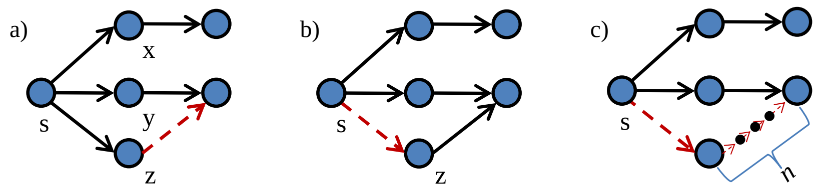

Fig. 2. Our recommendation to s is $-$ due to diminished trust in the $+$ voter. b) Our recommendation is 0 because z is untrusted, and hence we ignore z's outgoing edges. A system whose recommendation was influenced by z could be swayed arbitrarily by a careless or adversarial z.c) Here z is connected to the $+$ voter by a chain of distrust links. Again our recommendation to s is 0. The recommendation of systems obeying "the enemy of my enemy is my friend," would depend on whether n is even or odd.

w, then it is as if u trusts w directly, though the weight of the trust may be diminished. Matters are more complicated in the case of distrust, which is the focus of the paper.

The ancient proverb, "the enemy of my enemy is my friend," immediately comes to mind when considering distrust propagation. In our context, it suggests that if u distrusts v and v distrusts w, then u should trust w However, this principle is inappropriate in most multifaceted domains. For example, you would probably not necessarily select a doctor on the basis that the doctor is heavily distrusted by people you distrust. Figure 2 case (c) illustrates how this maxim may lead to a recommendation that depends on whether the number of nodes in a network is even or odd, an effect of questionable psychological plausibility. Analogously, "the friend of my enemy is my enemy" would suggest that in Figure 2 case (b), one should give a negative recommendation. However, in many cases such as the medical specialist domain, it seems harsh to criticize a doctor on the basis that someone you distrust also likes that doctor. Hence, the two principles that we do recognize are: "the friend of my friend is my friend," and, "the enemy of my friend is my enemy." Other principles which we adopt are that equal amounts of trust and distrust should cancel. These principles are formalized through axioms, described later. As mentioned, a random walk is a nice toy model of how one might ask for recommendations: you ask a random friend, and if he doesn't have an opinion, then he asks a random friend, and so forth. A second justification for these systems might be called an *equilibrium* justification based upon trust scores. Namely, imagine each node has a trust score $t_u \geq 0$, which indicates how trusted the node is. One should be able to justify these scores based on the network. One easy way to do this is to have the trust in an agent be the trust-weighted sum of the declared trust from other agents. Formally,

$$t_s = 1, t_u = \sum_v t_v \cdot w_{vu} \forall u \neq s, \text{and recommendation is sign}(\sum_{u \in V_+} t_u - \sum_{u \in V_-} t_u) \quad (1)$$

The trust scores represent self-consistent beliefs about how trusted each node should be.

1.2 Prior work on distrust propagation

We first argue why existing work fails to achieve our goals in the simple examples in Figure 2. Of course, each of these algorithms may have other appealing properties that our algorithm lacks, and may be more appropriate for other applications. In seminal work, Guha et al. [2] consider a number of different approaches to propagating distrust. None of them matches our behavior on simple examples like those of Figure 2. The first approach is to ignore distrust edges. (A similar approach is taken by the Eigentrust algorithm [5], where any node u which has more negative than positive interactions with a node v has the edge weight w_{uv} rounded up to 0). Of course, this approach fails to handle example 2a. Their second approach is to consider distrust at the end: after propagating trust only as in the first approach, a certain amount of distrust is "subtracted off," afterwards, from the nodes based on the steady-state trust-only scores. This is a reasonable and practical suggestion, and does agree with our behavior on the three examples in Figure 2. However, on the simple example in Figure 3 case (a), it would provide a neutral recommendation to s, since the distrust comes “too late,” after equal amounts of trust have already been propagated to both voters.

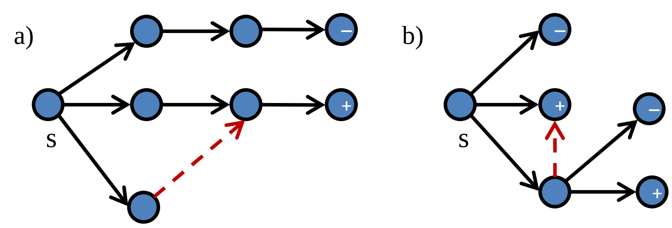

Fig. 3. Two further simple examples where our recommendation to s is $-$

The third approach of Guha et al. is to model distrust by negative weights so that $w_{vu} \in [-1, 1]$(now $\sum_u |w_{vu}| \leq 1$) and simply use the equations given above without change. This follows the “enemy of my enemy is my friend” dictim and hence has a parity dependence on n in Figure 2 case (c). It also gives a negative recommendation in Figure 2 case(b). Several others have adopted this “Plus-Minus Page Rank” approach. However, Guha et al. also consider more powerful notions of trust propagation via “co-citation,” which would be interesting to add into our setting.

The PageTrust system [6], considers a random walk which traverses trust edges but “remembers” the list of all distrusted nodes seen thus far, and makes sure never to visit such a node in the remainder of the walk. Such an algorithm would assign a neutral recommendation to node s in the example of Figure 3 case(b), because of the fact that nodes taking the middle path have necessarily not observed the distrust edge. The differences in recommendations between our system and theirs can be made more dramatic. While the random walk perspective is appealing, it is not obvious how to naturally extend it to distrust because it strongly depends on the order in which nodes are visited.

1.3 Our system

Adopting the “equilibrium trust score” view of the random walk system, we propose the following modification of the Page Rank equations:

$$t_s = 1; t_u = \max(0, \sum_v t_v \cdot w_{vu}), \forall u \neq s; R = \text{sgn}(\sum_{u \in V_+} t_u - \sum_{u \in V_-} t_u) \quad (2)$$

In particular, nobody can have a negative trust score: negative trust becomes 0 trust and hence a node that has a net negative incoming amount of trust is completely ignored: neither its vote nor its outgoing edges affect the system in any manner. This is consistent with our view that distrusted nodes are either to be viewed as adversarial or arbitrary, and they should be ignored. Determining which nodes should be distrusted is one challenge we face.

There are examples that show that the equations we propose need not have a unique solution. However, we show that the trust scores of the voters are unique and hence the recommendation of the system is unique. Note that in the case where $\sum_v |w_{uv}| < 1$ for each node (or equivalently where each node has a positive weight pointing back to s), a simple fixed-point argument implies a unique solution for trust scores. Also, if all weights are nonnegative, then it is straightforward to show that the voters have unique scores, by fixed-point theorems. In the case of trust only, the system is simply the random walk system of Andersen *et al.*, hence it is a proper generalization. In fact, it is somewhat surprising that no one has previously suggested this simple signed generalization of the Page Rank equations. The contributions of this paper are (1) introducing this simple modification and proving that it leads to a unique recommendation, (2) giving an efficient algorithm for computing recommendations, and (3) justifying the system using the axiomatic approach.

2 Notation and definitions

We follow the notation and definitions of Andersen *et al.* [1] when possible. Since we consider the slightly more general case of a weighted multigraph, a voting network is a partially labeled weighted graph. The nodes represent agents and edges represent either trust or distrust between the agents. A subset of nodes, called voters, is labeled with votes of + or −. The remaining nodes are nonvoters. The recommendation system must assign, to each source nonvoter, a recommendation in $\{-, 0, +\}$.

Definition 1. *A* **voting network** *is a directed, weighted graph* $G = (N, V_+, V_-, E)$ *where* N *is a set of nodes,* $V_+, V_- \subseteq N$ *are disjoint subsets of positive and negative voters,* E *is a multiset of weighted edges over the underlying set* $N \times N \times (\mathbb{R} \setminus \{0\})$. *(Parallel edges and self-loops are allowed.) For each node* u*, the total outgoing weight magnitude is at most 1,* $\sum_{e=(u,v,w)} |w| \leq 1$.

An edge $e = (u, v, w)$ from u to v, of weight $w > 0$, indicates that u allocates a w fraction of its *say* to trusting v, and we say u trusts v. If $w < 0$, then u allocates a $-w$ fraction of its *say* to distrusting v. Since we allow parallel edges, technically an agent may trust and distrust someone, and these may cancel as covered in Axiom 1. We also allow self-loops, i.e., *self-trust*, which is covered in Axiom 5. We say that edge (u, v, w) is a **trust (distrust) edge** if w is positive (negative). We say that there is a **trust path** from u to v if there is a sequence of edges $(u, a_1, w_1), (a_1, a_2, w_2), \ldots, (a_k, v, w_k) \in E$ such that $w_i > 0$. A **path** (or u **can reach** v) exists if such a sequence exists regardless of the sign of w_i.

We denote the weight of an edge by $\omega(e)$, so that $\omega(u, v, w) = w$. With slight abuse of notation, we also define $\omega_{uv} = \sum_{(u,v,w)\in E} w$ to be the **total weight** from u to v. A node u is a **sink** if $\omega_{uv} = 0$ for all $v \in N$, and u is a **partial sink** if $\sum_v |\omega_{uv}| < 1$.

A weight of 1 from u to v means that u places complete trust in v and only v, while a weight of -1 indicates that u's influence is completely invested in diminishing v's trustworthiness. The bound on the total absolute value weight leaving a node u bounds u's total (dis)trust expressed in the system. Note that we allow the total to be strictly less than 1. This allows an agent who, for example, does not know many people on the network, to express limited (dis)trust in some agents without expressing complete trust. How much node u trusts another node v is a relative measure which is defined with respect to how much u trusts other nodes. For each node u, the weights on the outgoing edges show the fraction of his trust that u puts on each of its neighbors via the outgoing edges. As a result the total sum of these fractions should not be more than 1.

Let $n = |N|$ be the number of nodes. We denote by $V = V_+ \cup V_-$ the set of **voters** and $V^c = N \setminus V$ the set of **nonvoters**. We write $(N, V_\pm, E)$ as syntactic shorthand for (N, V_+, V_-, E).

Definition 2. *A* **recommendation system** R *takes as input a voting network* G *and* **source** $s \in V^c$ *and outputs* **recommendation** $R(G, s) \in \{-, 0, +\}$.

We denote by $\text{sgn} : \mathbb{R} \to \{-, 0, +\}$ the function that computes the sign of its input. We denote by $(x)_+$ the nonnegative part of x, i.e., $(x)_+ = \max\{0, x\}$.

3 The system

In addition to outputting the final recommendation $r \in \{-, 0, +\}$, the system outputs further information $r_+, r_- \in [0, 1]$. These numbers indicate the fraction of personalized recommendations that were positive and negative. The summary $r = \text{sgn}(r_+ - r_-)$ may be sufficient information, but it may also be useful to know the magnitude of the recommendations. For example, a neutral recommendation $r = 0$ may be arrived at because essentially no trusted parties voted, or because $r_+ = r_- = 1/2$, meaning that all trust resolved in votes, half positive and half negative. This distinction may be useful. Note that $r_+ + r_- \leq 1$.

The main idea behind the system is to compute trust scores for each node u, $t_u \geq 0$. The source, s, is defined to have trust $t_s = 1$. For the remaining $u \neq s$, the following equation must be satisfied.

$$t_u = \left(\sum\nolimits_{v\in N} \omega_{vu} t_v\right)_+ \tag{3}$$

For real $z > 0$, recall that $(z)_+ = z$, and $(z)_+ = 0$ otherwise. A trust score of 0 means a node is mistrusted. The above equations determine a unique finite value t_v for any voter v (and more generally, any node v which

Input: $G = (N, V_\pm, E)$, $s \in V^c$.
Output: positive and negative fractions $r_+, r_- \in [0,1]$, and overall recommendation, $r \in \{-, 0, +\}$.
1. For each voter $v \in V$, remove all outgoing edges. Also remove all incoming edges to s. This makes $\omega_{vu} = 0$ and $\omega_{us} = 0$, for all $u \in N, v \in V$.
2. Remove all nodes $v \in N \setminus \{s\}$ which have no path to a voter, and remove the associated edges. That is, $N = \{s\} \cup V \cup \{u \in N \mid \exists \text{ path } e_1, \ldots, e_k \in E \text{ from } u \text{ to a voter } v\}$.
3. Solve for $x \in \mathbb{R}^N$: $x_u = 1 + \sum_v |\omega_{uv}| x_v$ for all $u \in N$.(x_u represents the expected number of steps until a walk halts when started at node u considering $|\omega_{uv}|$ as the probability of going from u to v.)
4. Solve the following linear program for t:

$$\text{minimize} \sum_u x_u \left(t_u - \sum_v \omega_{vu} t_v \right) \text{ subject to,} \tag{4}$$
$$t_s = 1$$
$$t_u \geq 0 \text{ for all } u \in N$$
$$t_u - \sum_v \omega_{vu} t_v \geq 0 \text{ for all } u \in N$$

5. Output $r_+ = \sum_{v \in V_+} t_v$, $r_- = \sum_{v \in V_-} t_v$, and final **recommendation** $r = \text{sgn}(r_+ - r_-)$.

Fig. 4. Abstract description of the system. Recall that $V = V_+ \cup V_-$ is the set of voters. Lemma 2 shows that (4) gives a solution to (3), and Lemma 1 shows uniqueness.

can reach a voter). The remainder of the nodes are irrelevant for the recommendation. The trust in positive voters is $r_+ = \sum_{v \in V_+} t_v$ and similarly $r_- = \sum_{v \in V_-} t_v$. The final recommendation is $r = \text{sgn}(r_+ - r_-)$.

In the case of both positive and negative weights, it is not obvious whether equations (3) have any solution, one solution, or possibly multiple solutions. Moreover, it is not clear how to compute such a solution. The remainder of this section shows that they do in fact have a unique solution, which the algorithm of Figure 4 computes. It is clear that the algorithm is polynomial-time because solving linear programs can be achieved in polynomial time.

Lemma 1. *Let $G = (N, V_\pm, w)$ be a weighted voting network in which there is a path from each node to a partial sink. Then there is a unique solution $t \in \mathbb{R}^N$ satisfying $t_s = 1$ and, for each $u \neq s$, $t_u = \left(\sum_v \omega_{vu} t_v\right)_+$.*

In order to prove Lemma 1, it is helpful to understand self-loops, $\omega_{uu} \neq 0$. Any self loop can be removed with very simple changes to the trust scores.

Observation 1 *Let N be a set of nodes, and let $\overline{w} \in \mathbb{R}^{N^2}$ be such that $\sum_v |\overline{w}_{uv}| \leq 1$ and $\overline{w}_{uu} \in (-1, 1)$ for each u. Let $w \in \mathbb{R}^{N^2}$ be defined so that $\omega_{uv} = \overline{w}_{uv}/(1 - \overline{w}_{uu})$ and $\omega_{uu} = 0$ for all $u \neq v$. Then $t \in \mathbb{R}^N$ satisfies $t_u = \left(\sum_v \omega_{vu} t_v\right)_+$ if and only if the vector $\bar{t} \in \mathbb{R}^N$, where $\bar{t}_u = t_u/(1 - \overline{w}_{uu})$ for each u, satisfies $\bar{t}_u = \left(\sum_v \overline{w}_{vu} \bar{t}_v\right)_+$.*

Proof. We have that,

$$t_u = \left(\sum_v \omega_{vu} t_v \right)_+ \Leftrightarrow$$
$$\bar{t}_u (1 - \overline{w}_{uu}) = \left(\sum_{v \neq u} \frac{\overline{w}_{vu}}{1 - \overline{w}_{vv}} \cdot \bar{t}_v (1 - \overline{w}_{vv}) \right)_+ \Leftrightarrow$$
$$\bar{t}_u (1 - \overline{w}_{uu}) = \left(-\overline{w}_{uu} \bar{t}_u + \sum_v \overline{w}_{vu} \bar{t}_v \right)_+ \Leftrightarrow$$
$$\bar{t}_u = \left(\sum_v \overline{w}_{vu} \bar{t}_v \right)_+ .$$

The last two equalities are equivalent by considering two cases: $\bar{t}_u = 0$ and $\bar{t}_u > 0$. The first case is trivial, and the second case follows from the fact that $\bar{t}_u > 0$ iff $\sum_{v \neq u} \overline{w}_{vu} \bar{t}_v > 0$.

Proof ((Lemma 1)). By Observation 1, all self loops may first be removed. The proof is by induction on $n = |N|$. In the case where $n = 1$ or $n = 2$, there is trivially a unique solution. Now, consider $n > 2$. Suppose for the sake of contradiction that there are two different solutions, t and t'. Consider three cases.

Case 1. There is some node u such that $t_u = t'_u = 0$. Imagine removing u (and its associated edges) from the graph. The graph will still have the property that there is a path from each node to a partial sink, because if such a path formerly passed through u, then the node linking to u is now a partial sink. By induction hypothesis, the resulting graph has a unique solution, $\bar{t}$. However, it is easy to see that the solutions t and t' restricted to $N \setminus \{u\}$ will both remain solutions to the equations of the lemma statement. This is a contradiction because t and t' agree on $N \setminus \{u\}$ and on u as well.

Case 2. There is some node $u \neq s$ such that $t_u > 0$ and $t'_u > 0$. Similarly, we will remove the node and apply the induction hypothesis. However, in this case, when we remove the node, we will propagate trust through the node as follows. We consider the graph $\overline{G}$ with nodes $\overline{N} = N \setminus \{u\}$ and weights $\overline{w}_{vw} = \omega_{vw} + \omega_{vu}\omega_{uw}$. Note that this transformation preserves $\sum_v |\overline{w}_{wv}| \leq 1$ but does not necessarily preserve $\omega_{ww} = 0$ for each w.

We now (tediously) argue that, in $\overline{G}$, every node can reach a partial sink. In G, consider a path $u_1, u_2, \ldots, u_k$ to a partial sink u_k. Now, one difficulty is that some edge which has $\omega_{u_i u_{i+1}}$ may have $\overline{w}_{u_i u_{i+1}} = 0$. However, if this happens, we claim that u_i must be a partial sink in $\overline{G}$. To see this, consider $|w| \in \mathbb{R}_+^N$ to be the vector where $|w|_v = |w_v|$ for all v. If we had used the weights $|w|$ instead of w in G, then after propagation it would still have been the case that the sum of the weights leaving u_i is at most 1. In the signed version, after propagation the weight magnitudes are only smaller, and the weight magnitude of $\omega_{u_i u_{i+1}}$ is strictly smaller, so u_i must be a partial sink. Hence, the "zeroing out" of edges does not create a problem in terms of removing a path from a node to partial sink. Similarly, if $u = u_i$ for some i, then either the path to the partial sink remains (deleting u_i) after propagating of trust or the node u_{i-1} must have become a partial sink.

We would like to apply the induction hypothesis to $\overline{G}$. However, this graph may have self-loops and therefor is not a valid voting network. By Observation 1, though, we can remove any self loops and change the solution to the equations of the lemma by a simple predictive scalar. Hence, by induction hypothesis, the resulting graph has a unique solution, $\bar{t}$. However, it is not difficult to see that the solutions t and t' restricted to $N \setminus \{u\}$ will both remain solutions to the equations of the lemma statement. This follows because the new equations of the lemma are simply the same as the old equations along with the substitution $t_u = \sum_v \omega_{vu} t_v$, which holds in both t and t' since both $t_u, t'_u > 0$. However, since they both agree with $\bar{t}$ on $v \neq u$ and satisfy $t_u = \sum_v \omega_{vu} t_v$, they must be the same which is a contradiction.

Case 3. For each node $u \neq s$, $\text{sgn}(t_u) \neq \text{sgn}(t'_u)$

Case 3a. There are at least two nodes $u, v \neq s$ for which $\text{sgn}(t_u) = \text{sgn}(t_v)$. The idea is to (carefully) merge nodes and use induction. WLOG, say $t_u, t_v > 0$ and $t'_u = t'_v = 0$. Now, we consider merging the two nodes into one node a. That is consider a new graph with node set $\overline{N} = \{a\} \cup N \setminus \{u, v\}$ and weights the same as in G except, $\overline{w}_{xa} = \omega_{xu} + \omega_{xv}$ for each $x \in N \setminus \{u, v\}$, $\overline{w}_{ax} = (t_u \omega_{ux} + t_v \omega_{vx})/(t_u + t_v)$. It is relatively easy to see that if u or v is a partial sink, then so is a. Consider the scores $\bar{t}, \bar{t}' \in \mathbb{R}^{\overline{N}}$ which are identical to t and t' except that $\bar{t}_a = t_u + t_v$ and $\bar{t}'_a = t'_u + t'_v = 0$. It is also relatively easy to see that the conditions of the lemma are satisfied by both of these scores. However, by induction hypothesis (again we must first remove self-loops, as above, using Observation 1), this again means that $\bar{t}' = \bar{t}$, which contradicts $\bar{t} \neq \bar{t}'$.

Case 3b. There are exactly three nodes s, u, v, and $t_u = t'_v = 0$, $t'_u, t_v > 0$. If we remove all the outgoing edges of u, t should still be a valid scoring for the new network. Since we have only 3 vertices, the only incoming edge to v is from s and as a result, t_v should be equal to ω_{sv}. With the same argument, we have $t'_u = \omega_{su}$. Since $t'_v, t_u = 0$, we can conclude that:

$$\omega_{su} \cdot \omega_{uv} + \omega_{sv} \leq 0$$
$$\omega_{sv} \cdot \omega_{vu} + \omega_{su} \leq 0$$

Since $|w_e| \leq 1$, it is not hard to argue that the only way to satisfy both these equations, is to have: $\omega_{su} = \omega_{sv} > 0$ and $\omega_{uv} = \omega_{vu} = -1$. But this means that u and v are not partial sinks and also they don't have a path to a partial sink since all their weight is pointed to the other one.

Lemma 2. *The solution to step 4 of the algorithm in Figure 4 is also a solution to eq. (3).*

Proof. We first claim that the equations in step 3 have a unique (finite) solution. To see this, consider a discrete random walk which, when at node u, proceeds to each node v with probability $|\omega_{uv}|$, and with probability $1-\sum_v |\omega_{uv}|$ halts. Hence, once the walk reaches a voter, it halts on the next step. The equations for x_u in step 3(a) represent the expected number of steps until the walk halts when started at node u. Hence there is a unique solution, and clearly $x_u \geq 1$. To see that x_u is not infinite, note that each node has a path to some voter. Let $w_{\min} = \min_{uv:\omega_{uv}\neq 0} |\omega_{uv}|$ denote the smallest nonzero weight. Then, starting at any node, after n steps, there is a probability of at least $w_{\min}^n$ of reaching a voter within n steps. Hence, the expected number of steps, i.e., "time to live," is at most $n/w_{\min}^n < \infty$.

Consider a third optimization problem.

$$\text{minimize} \sum_u x_u \left| t_u - \sum_v \omega_{vu} t_v \right| \text{ subject to:} \qquad t_s = 1, \quad \forall u: \ t_u \geq 0 \tag{5}$$

We first claim that the solution to the above optimization problem is also a solution to eq. (3). Suppose for contradiction that the solution to (5) has $t_u - (\sum_v \omega_{vu} t_v)_+ = \Delta \neq 0$ for some node u. Next we claim that changing t_u to $t_u - \Delta$ will decrease the objective by at least $|\Delta|$. The term in the objective corresponding to u decreases by $x_u|\Delta|$. (That term will become 0 unless $\sum_v \omega_{vu} t_v < 0$, but it will still decrease by Δ nonetheless.) The term in the objective corresponding to any other v can increase by at most $x_v|\omega_{uv}\Delta|$. Hence the total decrease is at least $|\Delta|(x_u - \sum_v x_v|\omega_{uv}|) = |\Delta|$, by definition of x_u.

Next, note that (5) is a relaxation of (4) in the sense that: (a) the feasible set of (5) contains the feasible set of (4), and (b) the objective functions are exactly the same on the feasible set of (4). Also notice that a solution to (3) is a feasible point for (4). Hence, the fact that the optimum of (5) is a solution to (3) implies that any solution to (4) is also a solution to (3).

The above two lemmas show that the output of the system is unique.

4 Axioms

The following axioms are imposed upon a recommendation system, R. Two edges, $e = (u, v, w)$ and $e' = (u, v, w')$ are *parallel* if they connect the same pair of nodes (in the same direction). The first axiom states that parallel edges can be merged, if they are the same sign, or *canceled* if they have opposite signs.

Axiom 1 (Cancel/merge parallel edges) *Let $G = (N, V_\pm, E)$ be a voting network. Let $e_1 = (u, v, w_1)$, $e_2 = (u, v, w_2)$ be parallel edges. If the two edges have opposite weights, $w_1 + w_2 = 0$, then the two edges can be removed without changing the recommendation of the system. Formally, let $G' = (N, V_\pm, E \setminus \{e_1, e_2\})$. Then $R(G, s) = R(G', s)$ for all $s \in N$. If the two edges have the same sign, then the two edges can be removed and replaced by a single edge of weight $w_1 + w_2$ without changing the recommendation of the system. Formally, let $G' = (N, V_\pm, \{(u, v, w_1 + w_2)\} \cup E \setminus \{e_1, e_2\})$. Then $R(G, s) = R(G', s)$ for all $s \in N$.*

It is worth noting that this axiom (and in fact almost all of our axioms), may be used *in reverse*. For example, rather than merging two edges of the same sign, a single edge may be *split* into two edges of the same sign and same total weight, without changing the system's recommendation.

Along these lines, an easy corollary of the above axiom is that, if $\omega(e_1) + \omega(e_2) \neq 0$, then the two edges can be merged into a single edge of weight $\omega(e_1) + \omega(e_2)$, regardless of sign.

For clarity of exposition, in the further axioms, we will make statements like " changing the graph in this way does not change the systems recommendation." Formally, this means that if one considered the different graph G' changed in the prescribed manner, then $R(G, s) = R(G', s)$.

Axiom 2 (Independence of irrelevant/distrusted stuff) *For voting network $G = (N, V_\pm, E)$,*

1. Let $s \in N$. Removing an incoming edge to s doesn't change the recommendation of the system for s. Similarly, removing outgoing edges from voters doesn't change the recommendation of the system.

2. Let $u \in N$ be a node which is not reachable from s by a path of trust edges. Then removing u (and its associated edges) doesn't change the recommendation of the system for s.

3. Let $u \in N$ be a nonvoter which has no path to a voter (through any type of edges). Then removing u (and its associated edges) doesn't change the recommendation of the system for s.

The first part simply states that it doesn't matter who trusts or distrusts the source, since the source is the one looking for a recommendation. Note that the second part implies if a node has only distrust edges pointing to it, then the node is ignored as it may be removed from the system. This crucial point can also be viewed as a statement about *manipulability.* In particular, if a node is in no manner trusted, it should not be able to influence the recommendation in any manner. For the third part, if a node and all of its trust can never reach a voter, then it is irrelevant whom the edge trusts.

Axiom 3 (Cancelation) *Consider the voting network $G = (N, V_\pm, E)$ with two edges e_1, e_2, one trust and one distrust, that terminate in the same node. Suppose that there is a constant $c \in \{-, 0, +\}$ such that for any of the following modifications to G, the system's recommendation is c: (1) the two edges are both removed; (2) the endpoint of two edges are both redirected to a new positive voter (with no other incoming edges) without changing any weights; (3) the endpoint of two edges are both redirected to a new negative voter (with no other incoming edges) without changing any weights; and (4) these two edges are both redirected to a new voter node (whose vote may be positive or negative), and the weights of edges are both* negated. *Then the recommendation of the system (without modification) is c.*

Note that the stronger the conditions on when a pair of edges may be canceled, the weaker (and better) the above axiom is. The above axiom states that, if a pair of edges wouldn't change the system's recommendation if they were directed towards a positive or negative voter, or negated, then they cancel or are at least irrelevant. In regards to this axiom, we also would like to mention the possibility of axiomatizing the system with real-valued output $r \in \mathbb{R}$ (we chose discrete recommendations for simplicity). In this latter case, the conditions under which a pair of edges is said to cancel would be even stronger, since the recommendation must not change by any amount, arbitrarily small. The same axiomatization we have given, with the real-valued variant of the last axiom, Axiom 6, again uniquely implies our recommendation system (now with final recommendation $r = r_+ - r_-$ instead of $r = \mathrm{sgn}(r_+ - r_-)$).

Axiom 4 (Trust Propagation) *Let $G = (N, V_\pm, E)$ be a network of trust and distrust edges. Consider nonvoter $u \neq s$ with no incoming distrust edges and no self-loop. Let $e = (u, v, w)$ be a trust edge $(w > 0)$. We can remove edge e and make the following modifications, without changing the recommendation $R(G, s)$:*

- *Renormalizaiton: replace each other edge leaving u, $e' = (u, v', w')$ by $(u, v', w'/(1 - w))$.*
- *Propagation: For each edge $e' = (x, u, w')$, replace this edge by two edges, $(x, u, w'(1-w))$ and $(x, v, w' \cdot w)$ directly from x to v.*

The above axiom, is somewhat subtle. The basic idea behind trust propagation is that an edge from u to v of weight $w > 0$ together with an edge from x to u of weight $w' > 0$ are equivalent to an edge from x to v of weight $w' \cdot w$, because x assigns a w' fraction of its trust in u and u assigns as w fraction of its trust in v. Andersen *et al.* [1] perform this propagation by removing edge (x, u, w) and adding in the direct edge $(x, v, w \cdot w')$ and, of course, when they remove (x, u, w) they must add a new edge for each outgoing edge from u (actually, their graphs are unweighted so it is slightly different).

The additional difficulty we face, however, is that u may have outgoing distrust edges, which we do not want to specify how to propagate. Instead, we *peel off* the trust going from u to v, and replace it directly by an edge from x to v. However, since the edge leaving u only accounts for a w fraction of the total possible trust assigned by u, the remaining edges must remain in tact and rescaled. While the above axiom is admittedly involved, it is satisfied by the Random Walk system of Andersen *et al.*, and is very much in line with the notion of propagation in a random walk. It has the advantage that it provides a description of how to (carefully) propagate trust in the midst of distrust edges. We also note that a simpler axiom is possible here, more in the spirit of Andersen *et al.*, in which trust-distrust pairs can be propagated (e.g., "an enemy of my friend is my enemy").

Axiom 5 (Self trust) *Let $G = (N, V_\pm, E)$ be a voting network. Let u be a nonvoter vertex in N and e be an edge from u to itself of weight $\omega(e) > 0$. Then e can be removed and the weights of all other edges leaving u scaled by a multiplicative factor of $1/(1 - \omega(e))$, without changing the recommendation of the system.*

Axiom 6 (Weighted majority) *Let $G = (N, V_\pm, E)$ be a star graph centered at $s \notin V$, $N = \{s\} \cup V$, with exactly $|V|$ edges, one trust edge from s to each voter, then the recommendation of the system is* $\mathrm{sgn}\left(\sum_{v \in V_+} \omega_{sv} - \sum_{v \in V_-} \omega_{sv}\right)$.

Note that the above axiom can be further decomposed into more primitive axioms, as is common in social choice theory. For example, Andersen *et al.* [1] use *Symmetry*, *Positive Response*, and *Consensus axioms*, which imply a majority property similar to the above.

5 Analysis

Lemma 3. *The system of Figure 4 satisfies Axioms 1–6.*

In order to prove Lemma 3, we need to show that Equation 3 or Figure 4 satisfies each of the Axioms 3–6. Because of the lack of space, we only present the proof for Axiom 3. The rest of the proofs can be found in the full version of the paper. In the rest of the proof we assume that $G = (N, V_\pm, E)$ is given and t_v is a valid score for v using Equation (3).

Cancelation: Consider $e_1 = (u_1, v, w_1)$ and $e_2 = (u_2, v, w_2)$ and call the new voter vertex z. First note that the trust scores obtained for any network does not depend on the sign of the votes by voters. As a result, if we direct e_1 and e_2 to a positive or a negative voter, we should obtain the same vector of scores, t', for all the vertices (by uniqueness). Let $G_- = (N, V_+, V_\cup\{z\}, E \cup \{(u_1, z, w_1), (u_2, z, w_2)\} \setminus \{e_1, e_2\})$ when z has a negative vote and G_+ defined the same way but z has a positive vote. It is easy to see that $t'_z = (w_1 \cdot t'_{u_1} + w_2 \cdot t'_{u_2})_+$. We consider 3 cases here:
Case 1. $w_1 \cdot t'_{u_1} + w_2 \cdot t'_{u_2} = 0$.
In this case, t' is a valid set of scores for G as well. By uniqueness, it means that $t = t'$ and as a result, $w_1 \cdot t_{u_1} + w_2 \cdot t_{u_2} = 0$. So if we remove e_1 and e_2,t is a valid solution to the new graph and the recommendation will not change.
Case 2. $w_1 \cdot t'_{u_1} + w_2 \cdot t'_{u_2} = p > 0$.
First, one can prove that, because $p > 0$, $R(G_-, s) \leq R(G, s) \leq R(G_+, s)$. The idea is as follows: consider the set of scores t' for G. It is easy to see that all vertices except v satisfy Equation 3. Define the infeasibility value of a vertex v for a given scoring t' by $\iota_{t'}(v) = |\left(\sum_{u \in N} \omega_{uv} t'_u\right)_+ - t'_v|$ and define the potential function $\phi(t') = \sum_{v \in N} \iota_{t'}(v)$.It is also easy to see that $\iota_{t'}(v) \leq p$. Now starting from t', we can reach to a feasible set of scores t as follows: Iteratively find v that has the maximum infeasibility value. Find a path from v to a sink vertex. Update the scores one by one from v to the sink vertex along this path. Call the new set of scores t. Initialize t to t' and update it as follows: First set the score of v such that $\iota_t(v) = 0$. Now, go to the next vertex along the path and based on the new scores of t update the score of the vertex such that its infeasibility value is set to 0. Note that during this score updating, $\phi(t)$ only decreases at each update. Also when we reach to a voter vertex u, if the score has been updating during this process and t'_u is changed to t_u, the sum of infeasibility values will be decreased by at least $|t_u - t'_u|$. As a result, after our iterative procedure converges, the change in the score of the voters $\sum_{v \in V_\pm} |t_v - t'_v| \leq p$ or in other words, it does not exceed the current infeasibility value over all vertices that is not more than p for t'. Now, the recommendation score for G_- is simply $\sum_{v \in V_+} t'_v - \sum_{v \in V_-} t'_v - p$ which is less than or equal to $\sum_{v \in V_+} t_v - \sum_{v \in V_-} t_v$(recommendation score for G) and that is less than or equal to $\sum_{v \in V_+} t'_v - \sum_{v \in V_-} t'_v + p$ which is the recommendation score for G_+. Since $R(G_-, s) = R(G_+, s)$, we should have $R(G_-, s) = R(G, s) = R(G_+, s)$. Now, consider removing e_1 and e_2 from G(call the new graph G'), first note that t' is a valid set of scores for G' as well. So the recommendation score for G' is $\sum_{v \in V_+} t'_v - \sum_{v \in V_-} t'_v - p \leq \sum_{v \in V_+} t'_v - \sum_{v \in V_-} t'_v \leq \sum_{v \in V_+} t'_v - \sum_{v \in V_-} t'_v + p$. As a result, $R(G_-, s) = R(G, s) = R(G', s) = R(G_+, s)$.
Case 3. $w_1 \cdot t'_{u_1} + w_2 \cdot t'_{u_2} = -p < 0$. Imagine negating the weights of e_1 and e_2 (the axiom is invariant to this operation). Note that t' is still a valid set of scores for the new graph and by uniqueness it is the only feasible set of scores. Now, we have $-w_1 \cdot t'_{u_1} - w_2 \cdot t'_{u_2} = p > 0$ and now, we can argue as above.

The main theorem is the following.

Theorem 1. *The system of Figure 4 is the unique system satisfying Axioms 1–6.*

Proof. Let G be a voting network and $s \in N$ be a node. It suffices to show that the axioms imply that there is at most one value for $R(G, s)$, since we know, by Lemma 3 that the system of Figure 4 satisfies the axioms. The idea of the proof is to apply a sequence of changes to the graph, none of which change the recommendation $R(G, s)$. The sequence will eventually result in star graph to which we can apply Axiom 6.

First, we simplify the graph by Axiom 2: eliminating all edges to s, all edges leaving voters, and all nodes that are not reachable by trust edges from s, as well as all nodes that have no path (trust/distrust) to any voter. We then apply axiom 5 to remove any trust self-loops. We finally apply axiom 1 to merge all parallel edges.

In the body of this proof (this part), we change the graph so that there are no distrust edges pointing to nonvoters. Lemma 4, following this proof, then implies the theorem. We proceed by induction on the total number of edges to nonvoters (trust or distrust), call this k. The induction hypothesis is that, if there are at most $k-1$ edges to nonvoters, then there is a unique recommendation for the system. Say $k \geq 1$. If there are no distrust edges between nonvoters, then we are done by Lemma 4. Otherwise, let (u, v) be a distrust edge between $u, v \in V^c$ (possibly $u = v$). If $v = s$, we can apply Axiom 2 to eliminate the edge, and then use the induction hypothesis. Now there must be at least one trust edge from a nonvoter, say a, to v, otherwise v would have been eliminated. Now, imagine running the system through the algorithm of Figure 4. The simplifications we have already executed mean that no edges or nodes will be removed during Steps 1 or 2. Step 4 assigns a unique trust score to each node in the system. Now, the plan is to eliminate either edge (a, v) or edge (u, v) (or both).

Consider three cases. **Case 1:** $t_a\omega_{av} + t_u\omega_{uv} = 0$. In this case, we will argue that Axiom 3 implies a unique recommendation. The reason is as follows. Consider any modification of the graph in which these two edges have been moved (and possibly negated) to a new voter. By induction hypothesis, since there are now $\leq k-1$ edges to nonvoters, the recommendation on this graph is unique and hence is equal to the recommendation given by the system of Figure 4. Also note that this system computes the unique solution to the equations (3). However, note that the same solution vector t satisfies equations in the modified graph, because moving both edges causes a change by an amount $t_a\omega_{av} + t_u\omega_{uv} = 0$ in the right hand side of any of these equations, regardless of where the edges are moved to, or even if they are both negated. Hence, the recommendation of the system is identical regardless of which voter they are moved to. Axiom 3 thus implies that there is a unique recommendation for the system.

Case 2: $t_a\omega_{av} + t_u\omega_{uv} > 0$. In this case, we can use Axiom 1 *in reverse* to split the edge (a, v) into two edges, one of weight $-t_u\omega_{uv}/t_a$ and the other of weight $\omega_{av} + t_u\omega_{uv}/t_a$. Now we can apply the same argument above on the edge (u, v) and the first of these two edges.

Case 3: $t_a\omega_{av} + t_u\omega_{uv} < 0$. In this case, we will split edge (u, v) into two, and move one of its part and edge (a, v) to a voter, thus again decreasing the total number of edges to non-voters by one. This time, we use Axiom 1 in reverse to split the edge (u, v) into two edges, one of weight $-t_a\omega_{av}/t_u$ and the other of weight $\omega_{uv} + t_a\omega_{av}/t_u$. Now, exactly as above, the pair of edges (a, v) and the (u, v) edge of weight $\omega_{uv} + t_a\omega_{av}/t_u$ exactly cancel in the system. Exactly as argued above, the recommendation of the system must be unique no matter which voter they are directed towards. Therefore, the recommendation of the system is unique.

Lemma 4. *Let $G = (N, V_\pm, E)$ be a voting network, and let $s \in N$. Suppose, further, that all negative edges point to voters, i.e., $\omega_{uv} \geq 0$ for all $u \in N$ and nonvoters $v \in V^c$. Let R be a recommendation system that satisfies Axioms 1–6. Then there is one unique possible value for $R(G, s)$.*

Proof. First apply Axiom 2 so that the source has no incoming links. Let $S = V^c \setminus \{s\}$ be the set of nonvoters, excluding the source. In the first part of the proof, we will simplify the graph so that there are *no links between members of S*, and that all negative edges point to voters. To do this, we will repeatedly apply propagation and self-propagation, as follows. Choose any ordering on the nonvoters (besides s), $\{u_1, u_2, \ldots, u_k\} = V^c \setminus \{s\}$. In turn, for each $i = 1, 2, \ldots, k$, we will remove all links from u_i to S. First, if u_i contains a self-loop, we apply Axiom 5 to remove it. Next, for $j = 1, 2, \ldots, k$, if there is a trust edge from u_i to u_j, then we remove it using Axiom 4. This will remove all outgoing trust edges from u_i to S. The key observation here is that once all trust edges are removed from u_i to S, further steps in the above iteration will not create any edges from u_i to S. Hence, after iteration $i = k$, there will be no edges between members of S. It remains the case that all negative edges point to voters.

The graph now has some number of trust links from s to S, as well as trust and distrust links from $V^c = S \cup \{s\}$ to V. We now propagate all trust links from S to V^c, using Axiom 4. As a result, the only edges are trust links from s to $N \setminus \{s\}$, distrust links from S to V, and distrust links from s to V. We further simplify by merging any parallel edges (Axiom 1). We now proceed to remove all distrust edges. First, consider any distrust edge from s to voter $v \in V$. Since we have merged parallel edges, there cannot be any trust edges from s to v, and we have already altered the graph so that there are no other trust edges to v. Hence, by Axiom 2, we can remove v (and the edge from s to v) entirely. Next, consider any distrust edge from $u \in S$ to $v \in V$. If there is no trust edge from s to v, then again by Axiom 2, we can remove v. If there is no trust edge from s to u, then we can remove u and edge (u, v). Otherwise, there is a distrust edge from u to v and trust edges from s to u and from s to v.

Consider three cases.

Case 1: $\omega_{sv} + \omega_{su}\omega_{uv} = 0$. Now we will completely eliminate the distrust edge. We apply Axiom 4 *in reverse* to the edge from s to v, to create a trust edge from u to v and increase the weight of the edge from s to u. Now, we merge the parallel edges from u to v using Axiom 1. A simple calculation shows that the trust and distrust will exactly cancel, and no edge will remain. Hence, we have eliminated the negative edge (u, v) without creating further negative edges.

Case 2: $\omega_{sv} + \omega_{su}\omega_{uv} > 0$. Then we apply Axiom 1 *in reverse* to split edge (s, v) into two edges, one of weight $-\omega_{su}\omega_{uv} > 0$ and one of weight $\omega_{sv} + \omega_{su}\omega_{uv} > 0$. We then proceed as in Case 1 to cancel the edge from u to v with the edge from s to u of weight $-\omega_{su}\omega_{uv}$.

Case 3: $\omega_{sv} + \omega_{su}\omega_{uv} < 0$. In this case, we eliminate the trust edge from s to v, as follows. We again apply Axiom 4 in reverse to the edge from s to v, again to create a trust edge from u to v and increase the weight of the edge from s to u. Then we merge the parallel edges from u to v using Axiom 1 (in fact, we must first split the negative edge from u to v into two parts so that one may cancel the trust edge). What remains is a distrust edge from u to v, and there is no longer any trust in v. Hence, we can finally remove the node v and the associated distrust edge. Thus, in all three cases, we were able to eliminate the distrust edge without creating a new distrust edge, maintaining the special structure of the graph.

After eliminating all distrust edges, we remain with trust edges from s to $S \cup V$. By Axiom 2, we can eliminate all edges in S and any edges in V that do not have trust coming from S. Finally, we can apply Axiom 6 to get that the recommendation of the system is unique.

6 Conclusion

In conclusion, we have suggested a simple set of principles to address trust and *distrust* in recommendation systems. A guiding principle, which is apparent in Axiom 2, is that of non-manipulability by untrusted agents. This is apparent in the design of our system and axioms, and also in the features we did not include. For example, it is also natural to consider a notion of *co-trust*, which may be described as follows. Consider two agents that trust the same people. They may be viewed as similar. If one of them then votes, say, positively then a system might be inclined to give a positive recommendation to the other [2]. However, such systems would tend to be manipulable in the sense that an adversarial agent, who is completed distrusted, could influence recommendations.

Without getting into interpretation of what distrust means (such as the difference between distrust and mistrust), the key beliefs we ascribe to are that distrust should be able to cancel with trust, and that distrusted agents should be ignored. Combined with trust propagation, this gives a unique, simple recommendation system.

References

1. R. Andersen, C. Borgs, J. Chayes, U. Feige, A. Flaxman, A. Kalai, V. Mirrokni, and M. Tennenholtz. Trust-based recommendation systems: an axiomatic approach. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pp. 199-208, 2008.
2. R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proc. Int. Conf. on World Wide Web*, pp. 403412, 2004.
3. T. Haveliwala. Topic-sensitive pagerank. In In *Proc. Int. Conf. on World Wide Web*, pp. 517526, 2002.
4. F. Heider, *The Psychology of Interpersonal Relations.* J. Wiley & Sons, New York, 1958.
5. S. Kamvar, M. Schlosser, and H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In *Proc. Int. Conf. on World Wide Web*, pp. 640651, 2003.
6. C. de Kerchove and P. Dooren. The PageTrust algorithm: how to rank web pages when negative links are allowed? In *Proc. SIAM Int. Conf. on Data Mining*, pp. 346352, 2008.
7. J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.
8. J. Kunegis, A. Lommatzsch, and C. Bauckhage. The slashdot zoo: mining a social network with negative edges. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pp. 741-750, 2009.
9. L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report, Stanford University, 1998.