SOFIA: Social Filtering for Robust Recommendations

Matteo Dell'Amico
Dipartimento di Informatica e Scienze dell'Informazione
Università di Genova
16146 Genova, Italy
dellamico@disi.unige.it

Licia Capra
Department of Computer Science
University College London
London, WC1E 6BT
United Kingdom
l.capra@cs.ucl.ac.uk

ABSTRACT

Digital content production and distribution has radically changed our business models. An unprecedented volume of supply is now on offer, whetted by the demand of millions of users from all over the world. Since users cannot be expected to browse through millions of different items to find what they might like, filtering has become a popular technique to connect supply and demand: *trusted* users are first identified, and their opinions are then used to create recommendations. In this domain, users' trustworthiness has been measured according to one of the following two criteria: *taste similarity* (i.e., "I trust those who agree with me"), or *social ties* (i.e., "I trust my friends, and the people that my friends trust"). The former criteria aims at identifying *competent* users, but is subject to abuse by malicious behaviours. The latter aims at detecting *well-intentioned* users, but fails to capture the natural subjectivity of tastes. We argue that, in order to be trusted, users must be *both* well-intentioned and competent. Based on this observation, we propose a novel approach that we call *social filtering*. We describe SOFIA, an algorithm realising this approach, and validate its performance on two real large-scale datasets. We demonstrate that the recommendations produced by SOFIA are both accurate and attack resilient.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*information filtering*; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*sociology*

General Terms

Algorithms, Security, Measurement

Keywords

Collaborative filtering, Social networks, Link analysis

1. INTRODUCTION

In his 2006 bestseller "The Long Tail" [1], Chris Anderson emphasizes how digital distribution has dramatically changed retailers' business models. Traditional retailers have a limited space they can use to stock items; market forces drive them to carry only those items that have the best chance to sell, thus losing less popular ones. With the advent of the Internet, retailers are not bound by the same physical constraints, so that more items can be offered. Moreover, consumers at a global level can now be easily reached, meaning that more items will be bought from the 'long tail'; this is a non negligible market: even though not many of each item in the tail are sold, the numbers add up to large volumes overall, given the length of the tail itself. As a result, while a traditional bookshop can hardly be expected to sell more than 100,000 different titles, an online service such as Amazon.com can offer its costumers millions of different products. More recently, not only the cost of distribution, but also the cost of production has been dramatically reduced; compare, for instance, the variety of programs offered by a traditional broadcast or cable TV, with what is on offer on sites like YouTube.com. The overall result is a society where an unprecedented volume of supply can meet the demand of millions of users from all over the world.

However, as Anderson points out, providing people with a massive choice is pointless, if that means they have to browse through thousands, or even millions, of potentially relevant items. Rather, people must be assisted in finding what they want. Filters can be used to *connect supply and demand*, making it easier for users to find the particular content that they would enjoy.

The most popular technique to realise this connection is collaborative filtering (CF) [9]. Most of the work on collaborative filtering has been focusing on identifying users with similar preferences, and then recommending items that people with similar tastes have approved. The adoption of collaborative filtering has arguably been a key factor in the success of Amazon.com: readers are invited to send reviews of books and to rate them on a five-star scale; other readers can comment on how useful the reviews were. By so doing, customers with similar tastes are detected and their past purchases used to create recommendations for similar users for what to buy next (i.e., thus connecting supply and demand). Traditional collaborative filtering techniques have worked quite well for the mass market and under the assumption of collaborative behaviours. However, these techniques have been subject to abuse by malicious behaviours [13]: for example, malicious users could copy honest users' reviews, to gain high similarity scores with them; they could subsequently inject inflated reviews in the system, to trick those users into buying an item or, viceversa, to disrupt an item's sales.

We argue that *accurate* and *robust* filtering techniques can be devised by exploiting information from a user's social network. We call this approach *social filtering*. The core

.

idea is to give higher weight to recommendations received from *trusted* users. To be trusted, a user must be both *well intentioned* and *competent*. Traditional collaborative filtering techniques focus only on competence (i.e., the ability to give useful - in a subjective way - recommendations), without considering the fact that competent users may indeed be malicious. Rather than relying on all recommendations from similar (i.e., competent) users, our approach specifically looks for well-intentioned users (i.e., users who are willing to provide honest recommendations) among those with whom we have stronger social relationships. Social ties are a warranty against malicious behaviors: if the trust inference algorithm is robust, it would be very costly for an attacker to build enough friendships with 'honest' users to effectively subvert the system. By so doing, we can prove that social filtering is more robust than traditional CF. Moreover, even in the absence of malign behaviour, we demonstrate that social filtering yields to more accurate recommendations, in situations where social ties intrinsically reveal information about users' similarity (e.g., two friends are more likely to share hobbies and tastes than two complete strangers); this is especially important when dealing with items that strongly appeal only to a small/niche community.

The remainder of the paper is structured as follows: Section 2 describes the philosophy behind social filtering, focusing on the two distinct aspects of intent and competence. In Section 3 we discuss SOFIA (SOcial FIltering Algorithm), that is, a specific realisation of social filtering. In Section 4 we analyse attacks against which filtering must defend itself, and in Section 5 we demonstrate the accuracy and robustness of SOFIA against two large real dataset, namely Citeseer and Last.fm. Finally, Section 6 concludes the paper and discusses future directions of research.

2. PHILOSOPHY OF THE APPROACH

Social filtering relies on the identification of *trusted* recommenders. In the scope of this work, we call trusted a recommender that is both well-intentioned and competent. The three questions we are thus trying to answer are: (1) how to evaluate intention (Section 2.1); (2) how to evaluate competence (Section 2.2); and (3) how to combine this information to find trusted recommenders (Section 2.3).

2.1 Intent

We define intent as the the *willingness* of a user to provide honest judgements[1]. Note that a judgement given with good intent is not necessarily useful, since users may have different tastes and preferences; Section 2.2 will illustrate how to find competent users among well-intentioned ones.

In our approach, a well-intentioned user is one that has a high *reputation*. We use the word 'reputation' here in its most general sense, that is, 'the estimation in which a person or object is held by the community or public' (source: Oxford Dictionary). Reputation is built over time: the more cooperative a user has been in the past, the higher their reputation. Note that high reputation is not developed by intrinsically honest users only: a selfish individual can be motivated to cooperate by sufficient incentives. If users benefit enough from being cooperative, reciprocative patterns emerge, and selfish individuals will maximize their payoff by cooperating [2, 18].

[1]In the following, we will use the more general term 'judgements', instead of 'recommendations', as our approach is equally applicable to recommendations (i.e., endorsements of products or content) as to 'negative' or purely informative judgements (e.g., "avoid that restaurant" or "this is relaxing music").

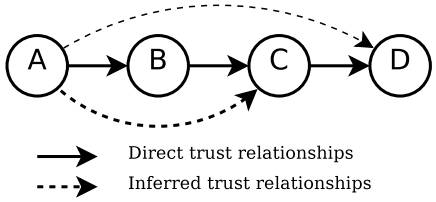


Figure 1: Transitive trust propagation pattern.

Finding users with high reputation is rather straightforward in centrally-managed systems: the system acts as a trusted third party, witnessing interactions and recording outcomes; if Alice (A) wishes to find out the reputation of Bob (B), A can simply query the system to get the answer she is looking for. Finding nodes with high reputation in distributed environments is not that trivial: interactions are not witnessed by trusted third parties, so it is not possible to keep global reputation records, as malicious users could report false outcomes of their interactions with other peers, in order to damage their reputation.

How can we find well-intentioned users in this case? Instead of global reputation, reciprocal *trust* relationships between nodes are maintained by the nodes themselves in the form of a *web-of-trust*, that is, a directed graph where nodes are users and an edge from A to B indicates that A trusts B (i.e., A has had direct interactions with B in the past and has reported she trusts B). A 'web-of-trust' is thus an instance of social network where ties between nodes are indeed based on 'trust' values. A problem arises when A has to judge the intentions of C, with whom she has never interacted before. In this case, it is sensible to give some trust to nodes that are recommended by nodes that we trust (and, iteratively, to nodes trusted by these new nodes as well). In other words, A *propagates trust over intent*, over the web-of-trust, from A itself to all nodes reachable from it via a directed path. It does so by means of the *transitive trust* propagation pattern shown in Figure 1. Usually, the level of trust inferred over a path is lower or equal to the minimum value of trust over the path; algorithms differ in how this quantity is calculated.

The principle of trust transitivity has been criticized since the judgement of who deserves trust is subjective [14, 10] (i.e., we are not guaranteed to like all the friends of our friends). However, we argue that, in the particular case of propagating trust over intent only (i.e., the willingness to provide honest judgements), transitivity is well justified: if node C has given honest judgements to node B (i.e., B trusts C), it is likely that C will do the same with node A too (A will trust C), as reciprocative behaviour creates an incentive for node C to be consistently well-behaved, thus building a high reputation [6]. Moreover, benevolent intent is a concept where subjectivity does not apply strongly, unlike competence, as discussed in the next section.

In this paper, we do not deal with social network creation and maintenance. Many different approaches could be used to accomplish this task, including: explicit social network creation (e.g., "Add as a friend" in sites like MySpace or

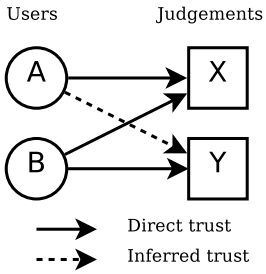


Figure 2: Co-citation trust propagation pattern.

FaceBook); use of email/phone-book contacts; automated creation as described in ReferralWeb [11]. Since the strength of our work lies in the combination of social network information and taste similarity, approaches that intrinsically create social networks purely based on taste similarity will experience the lowest gain from social filtering, as opposed to scenarios where the social network does bring in additional knowledge about nodes.

2.2 Competence

Together with intent, competence is a key component in evaluating the trustworthiness of recommenders. In this work, we define *competent* those users who are able to make correct judgments; since the definition of "correct" judgments is inherently subjective, competence is a subjective matter as well.

A sensible way of evaluating competence is via the co-citation pattern shown in Figure 2. A bipartite graph is used to represent a *network of judgments*: users and judgments form two disjoint sets of vertices (respectively $V_1 = \{A, B\}$ and $V_2 = \{X, Y\}$ in the graph); an edge (A, X) is present if user A expressed the judgment X. If users A and B agree on judgment X (i.e., there exist edges $A \rightarrow X$ and $B \rightarrow X$), then A may consider B a competent user from her viewpoint. Using the co-citation pattern, she may then *propagate trust over competence* on the judgement Y that B expressed.

However, users' competence is not sufficient to warrant trust to their judgements. For instance, let us consider a malicious user Mallory, wishing to trick Alice in believing a dishonest judgement Z stating that "Mallory's Greasy Restaurant offers very good food". In order to do so, Mallory could simply copy Alice's judgements; using the co-citation trust propagation pattern, Alice would deem Mallory a very competent evaluator, and would consequently believe/trust judgement Z too.

We argue that competence should thus be combined with intent to identify trustworthy recommenders, that is, recommenders who are willing to provide us with honest judgements and that we are likely to find useful.

2.3 The Combined Approach

As discussed above, using the *transitivity* trust propagation pattern alone is not enough, as subjectivity of tastes, which is an intrinsic characteristic of judgements, is lost. On the other hand, using the *co-citation* trust propagation pattern alone is subject to abuse by malicious users.

We propose a novel approach that combines the strengths of the two patterns, while circumventing their individual weaknesses: we exploit the transitivity trust propagation pattern on the web-of-trust to determine well intentioned

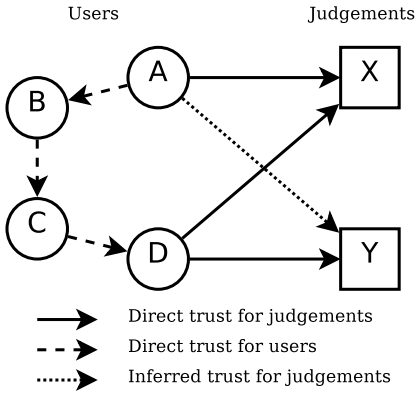


Figure 3: Combined trust-propagation approach.

users, and the co-citation trust propagation pattern on the network of judgements to evaluate their competence. By so doing, we are capable of inferring trust over judgements, in a way that is both accurate and robust. The underpinning idea is that, in order to be trusted, a judgement must have been expressed by a user who is both *willing* (intent) and *able* (competence) to give useful judgements. We call the new approach *social filtering*. Based on the interpretation of trust propagation over intent and competence we gave in the previous two sections, A can infer trust for a judgement Y expressed by a user D (Figure 3) if:

1. there exists a directed path from A to D in the web of trust (e.g., $A \rightarrow B \rightarrow C \rightarrow D$);
2. A and D both expressed at least one common judgement (e.g., X).

This is the first approach that aims at increasing the utility of recommendations, by exploiting information coming from the social network *and* from individual's preferences at the same time. We are aware of only two other works where the transitivity and co-citation trust propagation patterns have been used together, but with rather different goals and following a different philosophy: in [8], trust is propagated using *either* co-citation or transitivity in a social network where links represent similarity in preferences; in [16], the transitive trust propagation pattern is used as an *alternative to* the co-citation pattern, in order to bootstrap trust when traditional user similarity cannot be computed, again because of lack of information. These approaches work well in those scenarios where there is a strong correlation between social ties and individual preferences. On the contrary, our approach is best suited in those scenarios where the social network is not just a surrogate of users' preferences. As we shall demonstrate in Section 5, when separate information is available about the web-of-trust and judgements, an approach that reasons about intent and competence *at the same time* can yield the biggest increase in the utility of recommendations, even in the absence of malign behavior. Before doing so, we discuss how we have realised social filtering in practice.

3. REALIZATION OF THE APPROACH

In the previous section, we have introduced social filtering from a philosophical viewpoint, highlighting the advantages

of propagating trust over both intent and competence, in order to give users *trusted judgements*. To be of practical use, an implementation of social filtering would need to attribute a numeric value to the *amount* of trust a judgement deserves. This would ultimately allow users to rank judgements and/or to filter out unreliable ones. In this section, we first describe how the transitive and co-citation trust propagation patterns have been separately implemented in literature; we then motivate the choices we have adopted, and how they have been uniquely combined in SOFIA, our own implementation of social filtering. In describing our implementation, we will refer to the general case of weighted social networks, with weights expressing the strength of social ties. The user-judgement edges can be weighted as well, representing the level of confidence of a user towards a given judgement. The unweighted case is just a specific instance of the more general one, with all instances of trust relationships and/or judgements having the same weight.

3.1 Evaluating Intent

There exist various algorithms to quantify the amount of trust that is propagated transitively on a weighted social network. Properties that are common to most algorithms include:

- Longer paths disperse trust: if there is a trust path $A \rightarrow \ldots \rightarrow B \rightarrow C$, then the amount of trust inferred from A to C is not greater than the trust inferred from A to B.
- Adding paths increases trust: if there are two paths from A to B, then the trust that A infers for B is at least as high as if only one path was present.

Although intuitively sound, these properties alone are subject to Sybil attacks[2] [5]: in scenarios where new virtual identities can be cheaply created, a malicious node S_0 could create an unlimited number of siblings $S_1, S_2, \ldots$, add a web of strong (fake) ties between S_0 and its Sybil nodes S_i to the social network, and exploit this setup to gain a disproportionately large trust. To defend against this type of attack, trust propagation algorithms should limit the amount of trust gained by any Sybil node S_i by a function of the trust that S_0 has 'legitimately' gained.

Two popular approaches that guarantee this property are: the calculation of the maximal flow from the evaluator to the evaluated node, and the simulation of random walks on the web of trust. The use of the maximal flow function [6] guarantees that the inferred trust does not exceed the weight of each edge. Such metric has been shown to be "value-sybilproof" [3], that is, it is not possible for a single node S_i to obtain a trust value which is higher than the one that S_0 had before initiating the attack. Unfortunately, this result is relevant only when exactly one source node and one destination node are considered at any given time. However, in our scenario, the *overall* trust gained by the *set* of Sybil nodes must be limited too, as social filtering considers the opinions of many nodes simultaneously.

The idea of "random walks" has been exploited by PageRank [19], the algorithm used by Google for ranking search results. The algorithm considers a random walk over the graph of WWW pages and their links, starting from a random node and stopping with a probability $1 - \alpha$ at each step. Nodes are then ranked according to the probability that this random walk stops at them[3]. Pages that receive many incoming links, and pages that are being linked by another heavily-linked page, are then ranked higher. Intuitively speaking, the same approach could be used to propagate trust over a social network: the higher the number of paths (equivalent to links) leading to a node (equivalent to a WWW page), the more reputable the node is assumed to be (the higher it ranks).

The standard version of PageRank misses on subjectivity, as it ranks pages regardless of the evaluating node. As a consequence, any node in the system would propagate trust to a node X in the same way. To obtain a subjective version of the algorithm, two simple changes are required: first, we force the starting point of the random walk to be the evaluating node itself (thus avoiding walks that originate at malicious nodes); second, rather than having the same probability of jumping to another node (as done in the original version of PageRank), we chose such probability to be proportional to the weight (i.e., the strength) of the edge itself. A walk starting at A will thus result in trust propagation from A's subjective viewpoint only. This modified version of the original algorithm is sometimes referred to as *Personalised PageRank*.

The original version of PageRank is subject to Sybil attacks [7]. However, with Personalised PageRank, an attacker S_0 can only divert, towards the Sybil region, those paths that pass through S_0 itself. If the probability that a random walk reaches S_0 is p, then the cumulative value of all one-step paths from S_0 is αp; for two steps, it is $\alpha^2 p$, and so on. Thus, the maximal total rank for the Sybil region amounts to $\sum_{i=0}^{\infty} \alpha^i p = \frac{p}{1-\alpha}$. The α parameter thus influences the resilience to Sybil attacks: the lower the value of α the better the robustness. Low values of α also increase subjectivity, as they reward short paths over long ones, while when α approaches 1 the outcome of the algorithm becomes more and more similar, regardless of the initiator node. Finally, the lower the value of α the faster the convergence speed of the algorithm (with $\alpha = 0.5$, more than 99.9% of the overall ranking weight comes from paths of length up to 10). Note, however, that low values of α may cause honest nodes who are 'socially far-away' not to be considered, thus discarding potentially useful information. This may affect the accuracy of our algorithm, with respect to traditional collaborative filtering techniques where the full dataset is considered instead. We will analyse optimal choices of α with respect to accuracy vs. robustness in Section 5.

In our realisation of social filtering, we have chosen to deploy Personalised PageRank to quantify the transitive trust propagation over the social network, as it combines our requirements of subjectivity and robustness. A pseudo-code description of Personalised PageRank can be found in Algorithm 1. The computational complexity of the algorithm is proportional to the number of edges in the web of trust times the number of iterations needed to make the algorithm converge.

[2]This style of attack is also known as 'shilling' in recommender systems, 'profile injection' in collaborative filtering, and 'web spamming' in webpage ranking.

[3]The most common PageRank definition corresponds to the *equilibrium distribution* of a random walk, with a $1-\alpha$ probability of jumping to a random node. The two definitions are equivalent.

Algorithm 1 Personalised PageRank.

Parameters: a social network $G = (V, E)$; an evaluating node $A \in V$; weights such that w_{ij} is the weight of edge (i, j); a $0 < \alpha < 1$ parameter.
Returns: a vector r where r_i is the score of node i.
$n \Leftarrow$ size of V; $r \Leftarrow 0^n$; $r_A \Leftarrow 1$
while algorithm has not converged **do**
  $\hat{r} \Leftarrow 0^n$; $\hat{r}_A \Leftarrow 1 - \alpha$
  for all $i \in V$ **do**
    $d \Leftarrow \sum_{j \in V} w_{ij}$
    if $d > 0$ **then**
      for all j such that $(i, j) \in E$ **do**
        $\hat{r}_j \Leftarrow \hat{r}_j + \alpha \frac{w_{ij} r_i}{d}$
      end for
    else
      $\hat{r}_A \Leftarrow \hat{r}_A + \alpha r_i$ {If i is a sink we restart from A}
    end if
  end for
  $r \Leftarrow \hat{r}$
end while
return r

3.2 Evaluating Competence

The co-citation trust propagation pattern has been widely studied and applied to the problem of ranking Web pages. One of the most famous algorithms realising this pattern is HITS [12]. HITS conceptually divides pages in two subsets: authorities (i.e., pages whose content satisfy the query), and hubs (i.e., pages that link to relevant documents, that is, to authorities). Using an iterative process, HITS traverses the linkage structure of Web documents, and computes both a hub weight and an authority weight for each visited page at every step, so that:

1. Forward Step (from hubs to authorities): the weight given to an authority is proportional to the sum of the weights of those hubs linking to it;

2. Backward Step (from authorities to hubs): the weight given to a hub is proportional to the sum of the weights of those authorities being linked by it.

If weights expressing confidence are present in the network of judgements, they can be used as a multiplicative factor (i.e., a link with weight 2 acts as two separate links, each with weight 1). The process continues (renormalizing scores at every iteration) until it converges, and the top ranking pages, according to their authority scores, are then returned.

The principle behind HITS is that good hubs link good authorities, and good authorities are linked by good hubs, in a mutually reinforcing way. We argue that the same principle holds in our scenario, where we can expect competent users to give valuable judgements, and valuable judgements to be given by competent users. If we map users to hubs and judgements to authorities, we can run an HITS-like iterative algorithm to rank judgements, which is our ultimate goal. This would not realise our social filtering method though, as the following caveats must be addressed first.

(1) Solving the TKC Problem. It has been demonstrated that the HITS algorithm suffers from the "Tightly Knit Community" (TKC) syndrome [15]: if a community of users all gave the same (or very similar) judgements (thus resulting in a highly connected bipartite graph), the competence weight of the community would disproportionately increase, with the judgements they express being excessively high-ranked, even if they are not authoritative. A set of malicious users could thus artificially create a TKC in order to artificially boost their ranking.

To solve this problem, we adopt the solution proposed in SALSA [15]: we divide the weight that each hub transfers at each forward step by its outdegree (the sum of weights on outgoing edges), and we do the same for authorities and their indegree at each backward step. After a forward step, the total weight transferred from a single hub to its linked authorities is thus equal to the weight on that hub; vice-versa, after a backward step, the total weight that is redistributed from a single authority to the set of hubs linking to it equals the weight gained by the authority. Thus, the sum of weights remains constant at every step, removing the need for normalization.

A very desirable side-effect of this alteration is that users who express non-mainstream judgements are rewarded: the fewer the users who have expressed some judgements, the higher the weight that is transferred back on a per-user basis. In so doing, we express the fact that "niche" judgements are more significant than mainstream (redundant) ones: if a user, in her top 10 listenings, has both The Beatles and an unheard-of rock band, it is likely that the unknown rock band is more indicative of her musical preferences.

(2) Subjectivity of Ranking. HITS-like algorithms provide non-subjective results, as they are independent of the user A starting the search. To cater for the subjectivity required by our scenario, we initialize the algorithm so that the only hub (user) with a non-zero weight is the reference node A itself (instead of assigning an equal weight to any hub in the network). In so doing, the first forward step of the algorithm only considers the judgements given by the reference node, thus tailoring the ranking results to his/her tastes. To limit the propagation of trust to judgements that are too dissimilar from the tastes of A, after each backward step, the weights associated to each user are multiplied by a parameter $\beta \in (0, 1)$, and the trust given to A is increased by $1 - \beta$. These two changes are similar, in spirit, to the modifications already suggested for PageRank, where we forced the random walk to start from the very same node; the β parameter plays the same role that α plays in PageRank, ensuring the convergence of the algorithm, with lower values of β impling faster convergence and higher subjectivity.

(3) Catering for Well-Intentioned Users. As discussed in Section 2.2, trust propagation over competence alone is susceptible to attacks. We propose to add robustness to HITS-like algorithms, by incorporating users' intent assessment as follows. To begin with, Personalised PageRank (Algorithm 1) is run on the social network, thus obtaining a vector with nodes' reputation, as seen by the reference node A. We then run the subjective HITS-like algorithm, so that, at every backward step, trust is redistributed from judgements to users in a way that is *proportional to users' intent*, as measured by Personalised PageRank. In other words, *reputation becomes a multiplicative factor for backward trust propagation*. As discussed in Section 3.1, a Sybil coalition can obtain only a limited amount of trust from the social network, so the amount of trust that can be transferred to

Algorithm 2 SOFIA.

Parameters: a judgement bipartite network $G = (V, E)$, where V is the union of the set of users U and the set of judgements J; an evaluating node $A \in U$; weights such that w_{uj} is the weight of edge (u, j); an intent ranking vector r computed using Personalised PageRank over the web of trust, so that r_u is the intent ranking of user u; a $0 < \beta < 1$ parameter.
Returns: a trust vector $\hat{t}$ such that $\hat{t}_j$ is the trust ranking of judgement j.
$n \Leftarrow$ size of U; $m \Leftarrow$ size of J; $t \Leftarrow 0^n$; $t_A \Leftarrow 1$
while algorithm has not converged **do**
  {Forward Step: from users to judgements}
  $\hat{t} \Leftarrow 0^m$
  for all $(u, j) \in E$ **do**
    $\hat{t}_j \Leftarrow \hat{t}_j + \frac{w_{uj}}{\sum_{k \in J} w_{uk}} t_u$
  end for
  {Backward Step: from judgements to users}
  $t \Leftarrow 0^n$; $t_A \Leftarrow 1 - \beta$
  for all $(u, j) \in E$ **do**
    $t_u \Leftarrow t_u + \beta \frac{w_{uj} r_u}{\sum_{v \in U} w_{vj} r_v} \hat{t}_j$
  end for
end while
return $\hat{t}$

malicious nodes is limited too.

We call the algorithm that results from modifying the HITS-like approach in the three ways described above SOFIA, that is, SOcial FIltering Algorithm. The resulting pseudocode is shown in Algorithm 2.

The result of running SOFIA is a vector $\hat{t}$ containing a *trust* numeric value for each judgement in J, computed considering both the intent and the competence of the users in U, as seen by the reference node A. The normalization parameters $(\sum_{k \in J} w_{uk}, \sum_{v \in U} w_{vj} r_v)$ can be calculated outside the loops, so the computational cost of the algorithm is, similarly to PageRank, proportional to the number of edges in E times the number of iterations of the algorithm.

3.3 Example

To see how SOFIA works, let us consider the sample web of trust and judgement network depicted in Figure 4. The web of trust represents "friendship" relationships on a social networking site between users Alice (A), Bob (B), Carol (C) and Dave (D) (e.g., Alice listed Bob and Carol as friends; Bob and Dave mutually listed themselves as friends, etc.). Judgements represent recommendations of their favorite music bands: the Xenons (X), Yodels (Y), Zaars (Z) and Whistlers (W) (e.g., Alice, Bob and Carol all endorse the Xenons; Bob also likes Yodels, etc.).

We now want to give A, our reference user, recommendations about music bands she may like. We thus run SOFIA over the graphs displayed in Figure 4, where we assume all edges to have unitary weight. We also set the parameters $\alpha = \beta = 0.5$.

To begin with, Personalised PageRank is executed over the web-of-trust. The algorithm yields a normalised intent ranking of 0.50 for A, 0.231 for B, 0.154 for C and 0.115 for D. Note that B obtains a higher ranking than C, as he is

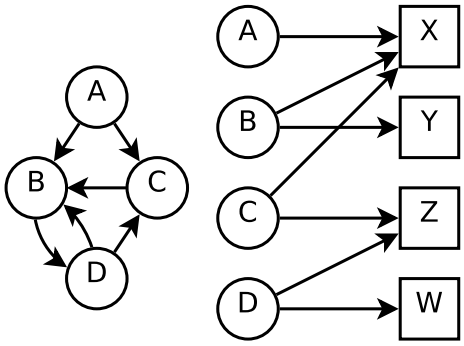


Figure 4: Example web of trust (left) and network of judgements (right).

trusted by all other users in the web of trust (despite both being at the same distance from A); D is penalized instead for being farthest from A.

After having computed user intent, we then run the core of the SOFIA algorithm, to rank judgements. The algorithm converges to vector $\hat{t} = \{0.909, 0.051, 0.035, 0.005\}$, listing the authority weights for X, Y, Z and W respectively. Note that Y obtains a higher ranking than Z: in fact, although they follow the same propagation pattern (one-step propagation from A via X), the intent ranking used during the backward step from X is higher for B than for C. For the opposite reason, W obtains the lowest ranking, as its recommendation is obtained via two steps of transitivity (via Z from C, and up via X from A), and it is endorsed by D, who has the lowest intent ranking.

4. ATTACK MODEL

In order to validate our social filtering algorithm, we have conducted a variety of experiments on two very large real datasets. While ideal to measure accuracy, real datasets are unsuitable to test the robustness of the algorithm while varying threat intensity. To demonstrate the robustness of SOFIA, we thus have to manually inject attacks on top of real datasets, and run experiments under different configuration settings. In this section, we analyse threat strategies, leaving their enactment and corresponding experimental validation to Section 5.

In the scenario we are considering, the most plausible goal of an attacker would be to alter the rating of a certain judgement X. It may do so either to trick a single user A, or more extensively to deviate the judgements of all users, in favour of (or against) X. Let us analyse how an attacker could achieve such goal. In the first case, since the attacker wants to be rated by A as a very competent user, it could first copy the judgements that A expressed, and then add a new judgement X. In the second case, there is no single set of judgements the attacker can copy, as each user would have expressed different ones: copying popular judgements would yield to very little reward, as a consequence of our strategy to reward users who gave niche judgements more (see Section 3.2 point (1)); on the contrary, copying 'niche' judgements would yield to very high appeal, but to rather few users. We will thus model this attack as we modeled the targeted attack, that is, by copying the judgements of a randomly chosen node A and adding the judgement for X; however, rather than studying the impact of the attack on

A, we will study the 'collateral damage' that the attack has on other users.

The attack strategies described above model the behaviour of one attacker only. However, to increase the impact of the attack itself (i.e., to increase the ranking of judgement X), we must also consider the case of an attacker who has the ability to create an unlimited number of Sybil identities, all endorsing X. We assume that each Sybil can create any number of outgoing edges in the web of trust, from the Sybil node to any other user. They can also create any number of incoming edges, originating within the Sybil coalition. However, what they cannot do is create incoming edges from honest nodes at will, since obtaining trust from well-intentioned peers is costly. It is thus reasonable to expect a low cut between the "honest" and the "Sybil" region [20]. In our experiments, we will thus create Sybil regions that are highly interconnected internally; we will then set the amount of incoming links from honest nodes as a parameter, and analyse the robustness of SOFIA (i.e., how highly ranked can X become) against it.

5. EXPERIMENTAL VALIDATION

We have evaluated SOFIA along two dimensions: accuracy and robustness against Sybil attacks. The results are reported in Section 5.2 and 5.3 respectively. Both experiments were conducted using data from two real datasets: the Citeseer online scientific digital library, and the Last.fm music and social networking website. The key characteristics of these datasets are briefly summarised below.

5.1 The Datasets

Citeseer (`http://citeseer.ist.psu.edu/oai.html`) is an online scientific literature digital library, containing over 750,000 documents. From this repository, we have extracted a social network based on the co-authorship relation: if A and B have co-authored n papers together, then an edge between the two will be added to the social network, with weight n. The judgement network is built from the citations instead: if a paper X authored by A cites paper Y, then an (unweighted) edge from A to Y is added to the judgement network; the rationale is that, by citing Y, the authors of X have expressed the judgement "Y is relevant with respect to the topic discussed in X". To obtain a more manageable subset of the whole network, we isolated a highly-clustered subset of 10,000 authors, and took in consideration only the papers that had them as authors. The result is a set of 182,675 different papers; 48,998 of them received at least one citation by one of the others.

Last.fm (`http://last.fm`) is a "social music" website that creates profiles of the musical taste of its users, by tracking which songs they listen more often to in their digital music players. As in other social networking websites, users can explicitly create an (unweighted) social network by adding other users to their friend-list. We gathered our social network with a breadth-first crawl of 10,000 users using the Audioscrobbler Web Services (`http://www.audioscrobbler.net/data/webservices/`). We then considered the 50 most listened artists of each user, and ended up with a total of 51,654 different artists. The judgement network was finally created by linking users to their most listened artists (thus representing the judgement "user A likes to listen to songs by X"), and by weighting each judgement edge with the number of times the user listened to songs by that artist.

5.2 Accuracy

To assess the accuracy of SOFIA in giving recommendations, we performed the following experiment on both datasets: we "hid" one random edge $A \rightarrow X$ from the judgement network, run SOFIA on the modified network, and use its output (i.e., a vector of weights) to rank all judgements from A's viewpoint; this is equivalent to producing recommendations, tailored to A, based on the computed ranking of judgements. Since X is a judgement that A expressed (before we hid it), A obviously approves of it, so a good recommendation engine should return X at a very high ranking. Thus, the highest the position of X in the ranked list of judgements, the better the accuracy of the ranking algorithm. In the Citeseer dataset, the experiment is equivalent to guessing a missing citation from a paper; in Last.fm, it means finding the missing artist in the top-50 chart of a user. In the following, all the results shown (for a given algorithm and set of parameter) were computed from 1,000 individual instances of the experiment.

The first set of experiments aimed at analysing the impact that the two different trust propagation patterns (transitivity and co-citation) individually had on prediction accuracy; at the same time, we wanted to quantify the effect that different choices of parameters had on it (namely α, β and the number of iterations). We thus separated the two "halves" of SOFIA into:

Personalised PageRank (PPR) : each user u is first ranked using PPR; the ranking r_u is then simply divided between all the judgements u has expressed (proportionally to the edge weight). PPR thus enables us to measure the impact of trust transitivity, while disregarding the network of judgements;

Non-SOcial FIltering Algorithm (N-SOFIA) : all nodes in the web-of-trust are given equal intent ranking, instead of relying on the Personalised PageRank output. N-SOFIA thus enables us to study the impact of the co-citation pattern while disregarding the social network.

The first parameter we have studied is the *number of iterations* needed to obtain satisfying results. Table 1 shows the percentiles of the ranking of the "hidden" judgements, when running both PPR and N-SOFIA on the Citeseer dataset, with α and β parameters chosen to optimize the results. As the table shows, a rather small number of iterations is enough to obtain very good results: for instance, after 10 iterations, 10% of the hidden judgements can be found in the top 2 returned results (i.e., recommendations) of PPR, and at the very top for N-SOFIA; half of the hidden judgements

Algorithm	Iterations	Ranking percentiles 10	50	75	90
PPR ($\alpha = 0.3$)	3	**2**	32	161	4293
	5	**2**	30	**115**	**1709**
	10	**2**	**29**	141	3341
N-SOFIA ($\beta = 0.05$)	3	**1**	12	67	**1060**
	5	**1**	12	**63**	1136
	10	**1**	**11**	72	1020

Table 1: Hidden judgement ranking of PPR and N-SOFIA (best results in bold) with different numbers of iterations on the Citeseer dataset.

Dataset	α	Ranking percentiles					
		5	10	25	50	75	90
Citeseer	0.2	**1**	**2**	**8**	33	132	3076
	0.3	**1**	**2**	**8**	**30**	**115**	**1709**
	0.85	2	4	11	48	242	3473
Last.fm	0.3	**5**	14	75	361	**2107**	**15064**
	0.5	**5**	**12**	**66**	**344**	2188	16025
	0.85	**5**	14	71	367	2289	15648

Table 2: Impact of α on hidden judgement ranking with Personalised PageRank.

Dataset	β	Ranking percentiles					
		5	10	25	50	75	90
Citeseer	0.02	**1**	**1**	**3**	14	87	2820
	0.05	**1**	**1**	**3**	**12**	**63**	**1136**
	0.3	**1**	**1**	4	17	93	1603
Citeseer (CF)	–	1	1	3	15	88	–
Last.fm	0.01	**2**	**6**	**32**	**157**	**822**	**3954**
	0.1	5	13	58	269	1305	10599
	0.3	8	20	89	404	1742	9878
Last.fm (CF)	–	3	8	36	204	1061	7735

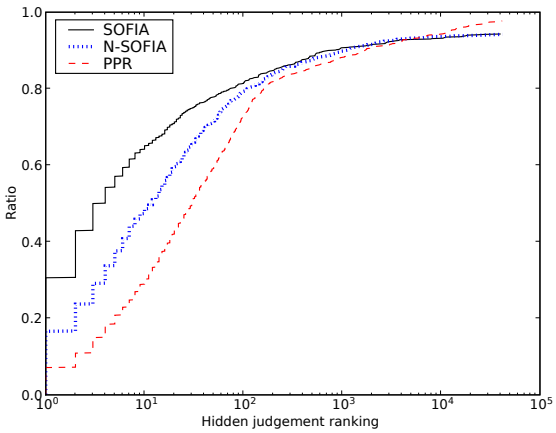
Table 3: Impact of β on hidden judgement ranking with N-SOFIA.

(50th percentile) were returned within the top 29 recommendations made by PPR, and in the top 11 by N-SOFIA, and so on[4]. Neither algorithms benefit from increasing the number of iterations after a small threshold. Rather, a higher number of iterations results in slightly worse results; our interpretation of this phenomenon is that, even if the weights on long paths are very low, they still introduce some "noise" since recommendations coming from longer paths are less reliable. In the following, the number of iterations for both parts of the algorithm has been set to 5.

We then studied the impact that parameters α and β had on the accuracy of PPR and N-SOFIA on the specific datasets at hand[5]. Table 2 and 3 report the results for different values of α on PPR, and of β on N-SOFIA, respectively. The key observation obtained from these numbers is that, on both datasets, N-SOFIA performs better than PageRank, suggesting that the information on tastes is more valuable than the information that can be inferred from the social network. On both datasets, the optimal value for β is much lower than the optimal value for α, suggesting that taste similarity propagates effectively on short paths only. Also, the optimal values for α are remarkably lower in our experiments than the "traditional" recommended $\alpha = 0.85$ for PageRank, reflecting the fact these datasets reward higher subjectivity. We have also compared the accuracy of N-SOFIA with traditional Collaborative Filtering techniques (in particular, using the cosine-based similarity measure): given that N-SOFIA produces recommendations based only on the network of judgements, while discarding social relations, we expect N-SOFIA and traditional CF to exhibit similar accuracy. As Table 3 illustrates (rows labeled CF), the accuracy is indeed comparable on both datasets. Note that attacks have not been considered yet: once introduced (Section 5.3), results will change dramatically, with approaches based on competence only (i.e., Collaborative Filtering-like techniques) suffering the most.

As a final set of experiments, we have compared the accuracy of PPR and N-SOFIA with SOFIA, under the best choice of parameters for both datasets. Results are shown in Figures 5 and 6, for Citeseer and Last.fm respectively (the graphs plot the cumulative distribution function for the ranking of hidden judgements). Using the Citeseer dataset, SOFIA outperforms both algorithms, with 50% of the hidden judgements being ranked in the top 4 positions, against 12 for N-SOFIA and 30 for PPR. The accuracy gain of SOFIA is perhaps more striking when considering up to 75% of the hidden judgements: using SOFIA, a user would find the hidden judgement in the the top-30 list of recommended papers, while using PPR the top-115 would have to be investigated. Of particular relevance is the observation that, even now that malicious attacks are *not* considered, SOFIA outperforms N-SOFIA, despite the fact that SOFIA throws away (potentially useful) information coming from (honest) socially far-away nodes. This means that SOFIA effectively exploits knowledge gathered from the social network to counter-balance this loss of data, and the gain is higher than the cost for datasets that, like Citeseer, exhibit the intrinsic property of having "socially close" nodes more likely to share tastes.

The performance gain of SOFIA on the Last.fm dataset is less striking. As Figure 6 demonstrates, SOFIA still outperforms PPR by a factor of 2. However, the performance

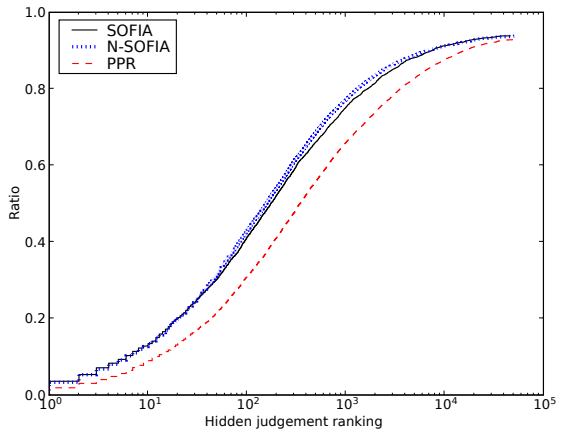


Algorithm	Ranking percentiles						
	5	10	25	50	75	90	95
SOFIA	**1**	**1**	**1**	**4**	**31**	**855**	–
N-SOFIA	**1**	**1**	3	12	63	1136	–
PPR	**1**	2	8	30	115	1709	**11609**

Figure 5: Hidden judgement ranking comparison on the Citeseer dataset. The α and β parameters were tuned for best performance ($\alpha = 0.5$, $\beta = 0.3$ for SOFIA, $\beta = 0.05$ for N-SOFIA, $\alpha = 0.3$ for PPR).

[4]Note that the judgements returned with ranking higher than of X are not mistakes: they are simply other recommendations that these algorithms compute but, given that such judgements were never made by A (unlike X), we have no way of measuring how accurate those are.

[5]Note that a single optimal choice of these parameters do not exist, as they intrinsically depend on the characteristics of the dataset (in terms of "level of transitivity").



Algorithm	Ranking percentiles 5	10	25	50	75	90	95
SOFIA	**2**	**6**	**32**	174	992	7429	–
SOFIA (2)	3	8	46	240	1347	11919	–
N-SOFIA	**2**	**6**	**32**	**157**	**822**	**6954**	–
PPR	5	12	66	344	2188	16025	–

Figure 6: Hidden judgement ranking comparison on the Last.fm dataset ($\alpha = 0.9$ and $\beta = 0.05$ for SOFIA, $\alpha = 0.5$ and $\beta = 0.1$ for SOFIA (2), $\beta = 0.01$ for N-SOFIA, $\alpha = 0.5$ for PPR).

of SOFIA and N-SOFIA are almost undistinguishable: with this dataset, the loss of data that SOFIA suffers from not considering far away nodes, and the added knowledge it gathers from the social network, balance each other out. However, even in these circumstances, we argue that running the whole SOFIA, instead of N-SOFIA alone, pays off: as we shall demonstrate in the next section, once attacks are in place, SOFIA outperforms N-SOFIA by far, thus yielding the best results overall in terms of accuracy *and* robustness. Note that the table in Figure 6 also reports the results of running SOFIA on an additional set of parameters, in particular, with a lower value of α; while accuracy becomes worse, we shall demonstrate, in the next section, that robustness to attacks becomes better, as shorter paths are considered, thus reducing the chance of traversing an attack region.

5.3 Robustness

As discussed in Section 4, we are interested in evaluating how much an attacker, with the ability of creating an unlimited number of fake nodes, can raise the ranking of a given judgement X. We assume that, while it is relatively cheap to create a fully connected Sybil sub-network, it is costly for any Sybil node to enter the social network of an honest node (i.e., to be directly trusted by a honest user). We have thus designed our experiments as follows: we have created a completely connected Sybil sub-network of 100 nodes, and attached it to the honest part of the web-of-trust with a parametric number k of *attack edges*; each attack edge is given a weight of 1, and the honest node to which it connects is chosen at random. All Sybil nodes copy all the judgements given by a random "victim" V, and then create another edge towards a malicious judgement X (in Last.fm, where judgements are weighted, the weight is set as the maximum between the judgements of the victim). We then study how the ranking of X changes, before and after the attack, both on V and on other random nodes in the network, *for different values of* k. Once again, for each algorithm and set of parameters, the results have been obtained with 1,000 instances of the experiment. Note that the strength of an attack on traditional CF techniques has usually been measured in terms of the proportion of malicious nodes in the network [17]; however, the number of Sybil nodes is not relevant for PPR- and SOFIA-like algorithms, where the impact of the attacker is limited by the total ranking of the Sybil region. We have thus fixed the number of Sybils to 100 in all experiments, while varying parameter k, which does influence the ranking of the Sybil region instead.

Table 4 shows how the ranking of malicious judgement X varies, with respect to parameter k, when enacting the attack on the Last.fm dataset (the results of the same experiment on the Citeseer dataset, not shown here for lack of space, are qualitatively equivalent, and all remarks expressed here are valid for both datasets). The α and β parameters were the same as those used for the experiments shown in Figure 6. The first row of the table shows the ranking of X when no attack is in place.

Let us consider N-SOFIA first. Since N-SOFIA does not take into account the social network, the number of attack edges k is irrelevant in this case. As shown, the malicious judgement X comes always at the very top of the recommendations made for the victim node V, even though, before the attack, such judgement was in position 12K or above! The ranking of X becomes very high even for nodes who are not specifically under attack, thus confirming the fact that N-SOFIA, like traditional collaborative filtering techniques based on taste similarity only, is highly vulnerable to Sybil attacks. Note also that, unlike PPR and SOFIA, algorithms that do not limit the amount of trust that a Sybil region can gain, will suffer more by (cheaply) increasing the number of Sybil nodes.

On the contrary, the impact of the attack on Personalised PageRank is marginal. In this case, being a victim is undistinguishable from being any node in the network, given that

Algorithm	k	Role	Percentiles 25	50	75
Any		no attack	12,914	25,827	38,741
N-SOFIA		victim	1	1	1
		other	348	1,185	3,132
PPR	1		10,730	20,493	33,322
	10		4,759	8,757	13,371
	100		1,092	2,012	3,101
SOFIA	1	victim	3,406	11,182	31,765
		other	9,599	19,186	33,064
	10	victim	469	1,311	2,815
		other	4,612	8,779	14,718
	100	victim	13	74	197
		other	1,040	2,649	5,571
SOFIA (2)	100	victim	138	353	697
		other	1,578	3,106	5,128

Table 4: Ranking of the "malicious judgement" after a Sybil attack, on the Last.fm dataset. k is the number of attack edges.

individual opinions are not taken into consideration. As the table shows, even when the Sybil region has conquered 100 attack edges, the ranking of the malicious judgement X is at position 2000 or above in 50% of the cases.

The robustness of SOFIA is comparable to that of PPR when considering non-victim nodes. The victim node clearly suffers instead, but much less than when using N-SOFIA: for example, when the Sybil region has 10 attack edges to the honest part of the network, 50% of the times the malicious judgement X is ranked at around position 1300 or above by the victim node using SOFIA, instead of position 1 using N-SOFIA. The impact of the attack becomes non-negligible for victim nodes running SOFIA once the number of attack edges reaches $k = 100$. Note, however, that this is a rather costly attack: in fact, it requires tricking 1% of the 10,000-node network into trusting dishonest nodes, and all this effort just to change the ranking of judgement X by a single node V, with X only gaining marginally in other nodes' viewpoints. This result supports the claim we made at the end of the previous section, that is, that running SOFIA pays off, as its accuracy is at least as good as that of N-SOFIA, but its robustness to Sybil attacks is ways higher. Last but not least, it is worth observing the impact of different choices of parameters on the robustness of SOFIA; the last set of results shown in Table 4 are obtained using the alternative set of parameters for SOFIA that were specified in Figure 6: while the accuracy of the recommendations using this second set of parameters was shown to be worse, the use of a lower α value makes the system more attack-resilient. As expected, there is a tradeoff between accuracy and robustness, and the desired balance between the two features can be obtained by adjusting the parameters to the specific characteristics and requirements of the domain at hand.

6. CONCLUSIONS

In this paper, we have proposed *social filtering*, a novel approach to realise accurate and robust recommendation systems, based on a combination of taste similarity and user intent. We have illustrated SOFIA, our realisation of social filtering, and demonstrated its accuracy against two real datasets, as well as its robustness against attacks of different magnitude. As shown, SOFIA achieves the best results in scenarios where judgements are subjective, and where users with similar tastes tend to form social ties.

We are currently extending the work presented in this paper in two directions. First, SOFIA is just an instance of social filtering. While we took inspiration from HITS and PageRank for the basic components of our algorithm, other approaches are certainly possible (for instance, using Pearson correlation or vector similarity to measure taste similarity). We are currently analysing the impact that different realisations of social filtering have on both accuracy and robustness. Second, we plan to work on a decentralized version of SOFIA that can be implemented on P2P and/or mobile systems. This would allow, for instance, the creation of effective and attack-resistant recommendations for content shared over those media. To obtain an efficient decentralized approximation, techniques that help finding short paths between nodes in a decentralized social network can be used [4].

Acknowledgments. Matteo Dell'Amico conducted this work while an intern at UCL, thanks to the financial support of the EPSRC under grant EP/E009190/1. The authors are also very grateful to Neal Lathia and Daniele Quercia, for their help with the experiments, the useful comments and the challenging discussions. "SOFIA" and the fictional rock band names are due to the creativity of Salvatore Scellato.

7. REFERENCES

[1] C. Anderson. *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion, 2006.

[2] R. Axelrod. *The Evolution of Cooperation*. Basic Books, New York, 1984.

[3] A. Cheng and E. Friedman. Sybilproof Reputation Mechanisms. In *3rd Workshop on Economics of Peer-to-Peer Systems, Philadelphia, PA.*, Aug. 2005.

[4] M. Dell'Amico. Mapping Small Worlds. In *7th IEEE International Conference on P2P Computing*, Sept. 2007.

[5] J. R. Douceur. The Sybil Attack. In *1st Intl. Workshop on Peer-to-Peer Systems (IPTPS '02)*, March 2002.

[6] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust Incentive Techniques for Peer-to-Peer Networks. In *ACM Conference on Electronic Commerce*, 2004.

[7] E. J. Friedman and A. Cheng. Manipulability of Pagerank under Sybil Strategies. In *Proceedings of the 1st Workshop of Networked Systems (NetEcon06)*, 2006.

[8] R. V. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of Trust and Distrust. In *Proc. of the 13th Intl. Conference on World Wide Web, (WWW 2004), New York, NY, USA, May 2004*, pages 403–412. ACM Press.

[9] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An Algorithmic Framework for Performing Collaborative Filtering. In *Proc. of the 22nd Intl. Conference on Research and Development in Information Retrieval (SIGIR '99)*, pages 230–237, New York, NY, USA, 1999. ACM Press.

[10] A. Josang. The Right Type of Trust for Distributed Systems. In *Proc. of the 1996 Workshop on New Security Paradigms*, pages 119–131, New York, NY, USA, 1996. ACM Press.

[11] H. Kautz, B. Selman, and M. Shah. Referral Web: Combining Social Networks and Collaborative Filtering. *Communications of the ACM*, 40(3):63–65, March 1997.

[12] J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *JACM: Journal of the ACM*, 46, 1999.

[13] S. Lam, D. Frankowski, and J. Riedl. Do You Trust Your Recommendations? An Exploration of Security and Privacy Issues in Recommender Systems. *Proc. of Emerging Trends in Information and Communication Security*, pages 14–29, Freiburg, Germany, 2006.

[14] M. Langheinrich. When Trust Does Not Compute – The Role of Trust in Ubiquitous Computing. Workshop on Privacy at Ubicomp 2003, Oct. 2003.

[15] R. Lempel and S. Moran. Salsa: the Stochastic Approach for Link-Structure Analysis. *ACM Transactions on Information Systems*, 19(2):131–160, April 2001.

[16] P. Massa and P. Avesani. Trust-aware Collaborative Filtering for Recommender Systems. In *International Conference on Cooperative Information Systems*, Cyprus, pages 492–508, October 2004.

[17] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams. Toward Trustworthy Recommender Systems: An analysis of Attack Models and Algorithm Robustness. *ACM Transactions on Internet Technology*, 7(4):23, 2007.

[18] M. A. Nowak and K. Sigmund. Evolution of Indirect Reciprocity by Image Scoring. *Nature*, 393(6685): 573–577, 1998.

[19] L. Page. PageRank: Bringing Order to the Web. Stanford Digital Libraries Working Paper 1997-0072, Stanford University, 1997.

[20] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: Defending Against Sybil Attacks via Social Networks. In *ACM SIGCOMM*, pages 267–278. 2006.