SOFIA: Social Filtering for Robust Recommendations

Matteo Dell'Amico and Licia Capra

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 criterion 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, in terms of accuracy and robustness, on two real large-scale datasets.

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 a limited number of items, in particular, those 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 a much wider variety of items can be offered from the 'long tail'. As a result, while a traditional bookshop can

Matteo Dell'Amico, Università di Genova, Italy e-mail: dellamico@disi.unige.it
Licia Capra, University College London, UK e-mail: l.capra@cs.ucl.ac.uk

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. 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) [7]. 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. 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 [11]: 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. Indeed, the robustness of CF systems is usually measured in terms of the proportion of malicious nodes in the network, under the assumptions that attackers are not able to create unlimited new identities at will, and they are not aware of the judgements expressed by each peer [18, 2, 16, 15]. In our approach, these assumptions can be dropped, and the impact of an attack becomes limited by the "intent" ranking of the attacker, which is in turn determined only by the connectivity of malicious nodes in the social network.

The remainder of the paper is structured as follows: Section 2 describes the concept motivating 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.

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; (2) how to evaluate competence; and (3) how to combine this information to find trusted recommenders.

Intent - Trust over Users
We define intent as the the *willingness* of a user to provide honest judgements[1], differentiating "spammers" from people who are legitimately using the application. Note that a judgement given with good intent is not necessarily useful, since users may have different tastes and preferences; this section will illustrate how to find competent users among well-intentioned ones.

Users' intent can be represented as a *web of trust*, that is, as a a directed graph where nodes are users and an edge from user *A* to *B* indicates that *A* considers *B* a well-intentioned one; in other words, *A* trusts *B*. Webs of trust are thus instances of social networks where links represent assessments on the behaviour of nodes rather than simple acquaintance.

The web of trust can be built in many different ways. For example, by means of explicit social network creation (e.g., "Add as a friend" in sites like MySpace or FaceBook); using email/phone-book contacts; via automated creation as described in ReferralWeb [9], and so on. We are not concerned with what specific technique is used to create the web of trust; however, we expect it to be difficult, for malicious nodes, to obtain endorsements from honest ones: this condition is key for the robustness of social filtering. For this reason, we discourage the creation of the web of trust via automated matching purely based on users' similarity.

The web of trust can then be traversed in order to obtain *reputation*[2] information about users we do not directly know and trust. We propose to do so by means of the *transitive trust propagation pattern*. A peer *A* obviously trusts the nodes that can be reached from itself via an edge; since *A* believes these nodes behave honestly, their recommendations for other nodes are believed by *A* to some extent, and some trust is propagated to them. The pattern repeats iteratively, propagating trust to all nodes reachable with a directed path starting from *A*.

The principle of trust transitivity has been criticized since the judgement of who deserves trust is subjective [12, 8] (i.e., we are not guaranteed to like all the friends of our friends). However, we argue that benevolent intent (unlike competence) is a concept where subjectivity does not apply strongly. Moreover, if the web of trust is

[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").

[2] 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)

built using evaluations of past behavior, reputation provides incentives to cooperation via reciprocative behavior [17, 4].

Competence - Trust over Judgements

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 so called *co-citation pattern*. A bipartite graph is used to represent a *network of judgments*: users (e.g., $\{A,B\}$) and judgments (e.g., $\{X,Y\}$) form two disjoint sets of vertices; 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. Using the co-citation pattern, she may then *propagate trust over competence* on the other judgements 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.

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 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 (Fig. 1) 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).

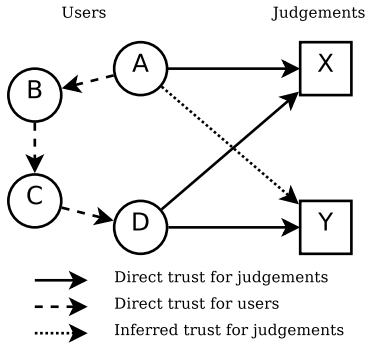

Fig. 1: Combined trust-propagation approach.

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 [6], trust is propagated using *either* co-citation or transitivity in a social network where links represent similarity in preferences; in [14], 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 to 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 conceptual 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 describe how the transitive and co-citation patterns 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.

Evaluating Intent

There exist various algorithms to quantify the amount of trust that is propagated transitively on a weighted social network. Desirable properties that most algorithms guarantee are: *longer paths disperse trust* (i.e., 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* (i.e., 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).

A popular approach that guarantees these properties is the simulation of a random walk on the web of trust, as done 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* (PPR).

Note that the original version of PageRank is subject to Sybil attacks[4] [5, 3]: 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

[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.

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

by a function of the trust that S_0 has 'legitimately' gained. Personalised PageRank does exactly so: 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.

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 [10]. 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 [13]: 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 [13]: 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; viceversa, 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 "niche" judgements are rewarded more than those expressing only mainstream (redundant) ones.

(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, 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 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 PPR. In other words, *reputation becomes a multiplicative factor for backward trust propagation*. As discussed in Section 3, 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 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 1. The result of running SOFIA is a vector

Algorithm 1 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 $\hat{t}$

$\hat{t}$ 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 full algorithm is proportional to the number of edges in E times the number of iterations of the algorithm.

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; 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.

To increase the impact of the attack itself (i.e., to increase the ranking of judgement X), we 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. 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.

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 musical tastes, by tracking which songs users listen more often to. Users explic-

itly create a 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 available at `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.

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 used 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 PPR 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 (50th percentile) were returned within the top 29 recommendations

Algorithm	Iterations	Ranking percentiles 5	10	25	50	75	90	95
PPR ($\alpha = 0.3$)	3	**1**	**2**	8	32	161	4,293	–
	5	**1**	**2**	8	30	**115**	**1,709**	**11,609**
	10	**1**	**2**	**7**	**29**	141	3,341	20,287
N-SOFIA ($\beta = 0.05$)	3	**1**	**1**	**3**	12	67	**1,060**	–
	5	**1**	**1**	**3**	12	**63**	1,136	–
	10	**1**	**1**	**3**	**11**	72	1,020	–

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

made by PPR, and in the top 11 by N-SOFIA, and so on[5]. 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[6]. Tables 2a and 2b 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 PPR, 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 mea-

Dataset	α	Ranking percentiles 5	10	25	50	75	90
Citeseer	0.2	**1**	**2**	**8**	33	132	3,076
	0.3	**1**	**2**	**8**	**30**	**115**	**1,709**
	0.85	2	4	11	48	242	3,473
Last.fm	0.3	**5**	14	75	361	**2,107**	**15,064**
	0.5	**5**	**12**	**66**	**344**	2,188	16,025
	0.85	**5**	14	71	367	2,289	15,648

(a)

Dataset	β	Ranking percentiles 5	10	25	50	75	90
Citeseer	0.02	**1**	**1**	**3**	14	87	2,820
	0.05	**1**	**1**	**3**	**12**	**63**	**1,136**
	0.3	**1**	**1**	4	17	93	1,603
Citeseer (CF)	–	1	1	3	15	88	–
Last.fm	0.01	**2**	**6**	**32**	**157**	**822**	**3,954**
	0.1	5	13	58	269	1,305	10,599
	0.3	8	20	89	404	1,742	9,878
Last.fm (CF)	–	3	8	36	204	1,061	7,735

(b)

Table 2: (a) Impact of α on hidden judgement ranking with Personalised PageRank. (b) Impact of β on hidden judgement ranking with N-SOFIA.

[5] 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.

[6] 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	**1**	**1**	**1**	**4**	**31**	**855**	–
N-SOFIA	**1**	**1**	3	12	63	1,136	–
PPR	**1**	2	8	30	115	1,709	**11,609**

(a)

Algorithm	Ranking percentiles						
	5	10	25	50	75	90	95
SOFIA	**2**	**6**	**32**	174	992	7,429	–
SOFIA (2)	3	8	46	240	1,347	11,919	–
N-SOFIA	**2**	**6**	**32**	**157**	**822**	**6,954**	–
PPR	5	12	66	344	2,188	16,025	–

(b)

Table 3: (a) 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). (b) 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).

sure): 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 2b illustrates (rows labeled CF), the accuracy is indeed comparable on both datasets. Note that attacks have not been considered yet: once introduced, results will change dramatically, with approaches based on competence only (i.e., CF-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 Tables 3a and 3b, for Citeseer and Last.fm respectively. 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 Table 3b demonstrates, SOFIA still outperforms PPR by a factor of 2. However, the performance 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 Table 3b also reports the results of running SOFIA on an additional set of parameters (row labelled SOFIA

(2)), 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.

Robustness

As discussed in Section 4, we are interested in evaluating how much an attacker, with the ability of creating an unlimited number of Sybils, 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 an 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 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, while varying 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 Table 3b. The first row of the table shows the ranking of X when no attack is in place.

Let us consider N-SOFIA and cosine-based collaborative filtering (CF) first. Since these algorithms do not take into account the social network, the number of attack edges k is irrelevant in these cases. 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 2.5K or above! The ranking of X becomes very high even for nodes who are not specifically under attack, thus confirming the fact that both N-SOFIA and traditional collaborative filtering techniques based on taste similarity only are highly vulnerable to Sybil attacks. On the contrary, the impact of the attack on PPR is marginal. In this case, being a victim is undistinguishable from being any node in the network, given that 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.

Algorithm	k	Role	Percentiles						
			5	10	25	50	75	90	95
Any		no attack	2,583	5,165	12,914	25,827	38,741	46,489	49,071
N-SOFIA		victim	1	1	1	1	1	1	1
		other	34	85	348	1,185	3,132	5,875	7,482
CF		victim	1	1	1	1	1	1	1
		other	25	54	214	1,522	27,367	–	–
PPR	1		2,297	4,459	10,730	20,493	33,322	–	–
	10		1,285	2,353	4,759	8,757	13,371	19,648	26,846
	100		334	559	1,092	2,012	3,101	4,434	5,290
SOFIA	1	victim	679	1,386	3,406	11,182	31,765	–	–
		other	2,264	4,409	9,599	19,186	33,064	–	–
	10	victim	41	132	469	1,311	2,815	7,039	34,725
		other	1,082	2,126	4,612	8,779	14,718	22,254	26,959
	100	victim	1	2	13	74	197	377	564
		other	215	391	1,040	2,649	5,571	8,395	10,179
SOFIA (2)	100	victim	15	46	138	353	697	1,042	1,234
		other	448	705	1,578	3,106	5,128	7,447	9,187

Table 4: Ranking of the "malicious judgement" after a Sybil attack on the Last.fm dataset.

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 way 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 Table 3b: 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.

References

1. C. Anderson. *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion, 2006.
2. R. Burke, B. Mobasher, R. Bhaumik, and C. Williams. Segment-based injection attacks against collaborative filtering recommender systems. In *Proc. IEEE ICDM '05*, 2005.
3. J. R. Douceur. The Sybil attack. In *Proc. IPTPS '02*, March 2002.
4. M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust incentive techniques for peer-to-peer networks. In *Proc. ACM EC '04*, 2004.
5. E. J. Friedman and A. Cheng. Manipulability of pagerank under sybil strategies. In *Proc. NetEcon06*, 2006.
6. R. V. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *ACM WWW '04*, 2004.
7. J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proc. ACM SIGIR '99*, 1999.
8. A. Josang. The right type of trust for distributed systems. Proc. *Proc. ACM NSPW '96*, 1996.
9. H. Kautz, B. Selman, and M. Shah. Referral web: combining social networks and collaborative filtering. *Commun. ACM*, 40(3):63–65, March 1997.
10. J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *JACM*, 46, 1999.
11. S. Lam, D. Frankowski, and J. Riedl. Do You Trust Your Recommendations? An Exploration of Security and Privacy Issues in Recommender Systems. In *Proc. ETRICS '06*, 2006.
12. M. Langheinrich. When trust does not compute – the role of trust in ubiquitous computing. Workshop on Privacy at Ubicomp 2003, October 2003.
13. R. Lempel and S. Moran. Salsa: the stochastic approach for link-structure analysis. *ACM Trans. Inf. Syst.*, 19(2):131–160, April 2001.
14. P. Massa and P. Avesani. Trust-aware recommender systems. In *Proc. ACM RecSys '07*, 2007.
15. B. Mobasher, R. Burke, R. Bhaumik, and C. Williams. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM TOIT*, 7(4):23, 2007.
16. B. Mobasher, R. Burke, and J. J. Sandvig. Model-based collaborative filtering as a defense against profile injection attacks. In *Proc. AAAI'06*, 2006.
17. M. A. Nowak and K. Sigmund. Evolution of indirect reciprocity by image scoring. *Nature*, 393(6685):573–577, 1998.
18. M. O'Mahony, N. Hurley, and G. Silvestre. Promoting Recommendations: An Attack on Collaborative Filtering. In *Database and Expert Systems Applications*. Springer, 2002.
19. L. Page, S. Brin, R. Motwani and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Digital Libraries Project 1999-66, Stanford University, 1999.
20. H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: defending against Sybil attacks via social networks. In *ACM SIGCOMM 2006*, pages 267–278. ACM, 2006.