Information Sciences 180 (2010) 95–112

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

The SocialTrust framework for trusted social information management: Architecture and algorithms

James Caverlee [a,*], Ling Liu [b], Steve Webb [b]

[a] Department of Computer Science, Texas A&M University, TAMU 3112, College Station, TX 77843-3112, USA
[b] College of Computing, Georgia Tech Atlanta, GA 30332-0765, USA

ARTICLE INFO

Article history:
Received 2 October 2008
Received in revised form 31 March 2009
Accepted 24 June 2009

Keywords:
Trust
Social web
Reputation
Relationship quality

ABSTRACT

Social information systems are a promising new paradigm for large-scale distributed information management, as evidenced by the success of large-scale information sharing communities, social media sites, and web-based social networks. But the increasing reliance on these social systems also places individuals and their computer systems at risk, creating opportunities for malicious participants to exploit the tight social fabric of these networks. With these problems in mind, this manuscript presents the SocialTrust framework for enabling trusted social information management in Internet-scale social information systems. Concretely, we study online social networks, consider a number of vulnerabilities inherent in online social networks, and introduce the SocialTrust framework for supporting tamper-resilient trust establishment. We study three key factors for trust establishment in online social networks – trust group feedback, distinguishing the user's relationship quality from trust, and tracking user behavior – and describe a principled approach for assessing each component. In addition to the SocialTrust framework, which provides a network-wide perspective on the trust of all users, we describe a personalized extension called mySocialTrust, which provides a user-centric trust perspective that can be optimized for individual users within the network. Finally, we experimentally evaluate the SocialTrust framework using real online social networking data consisting of millions of MySpace profiles and relationships. While other trust aggregation approaches have been developed and implemented by others, we note that it is rare to find such a large-scale experimental evaluation that carefully considers the important factors impacting the trust framework. We find that SocialTrust supports robust trust establishment even in the presence of large-scale collusion by malicious participants.

1. Introduction

The explosive rise of collaborative online communities has had a profound transformative impact on knowledge discovery and dissemination, business strategy and commerce, and even the structure of our interpersonal relationships. Community-powered services like the online encyclopedia Wikipedia, the image-sharing site Flickr, the community news portal Digg, and the video-sharing site YouTube have flourished as users have explored and contributed to these community-based information spaces. Similarly, social networking services like the ones offered by MySpace and Facebook have encouraged large-scale relationship management and new forms of collective information sharing.

Web-based social networks, online social media sites, and large-scale information sharing communities are prominent examples of *Social Information Systems*. These new systems are emerging as a promising new paradigm for large-scale

* Corresponding author. Tel.: +1 979 209 9998 (office); fax: +1 979 847 8578.
E-mail addresses: caverlee@cs.tamu.edu (J. Caverlee), lingliu@cc.gatech.edu (L. Liu), webb@cc.gatech.edu (S. Webb).

0020-0255/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2009.06.027

distributed data management and collective intelligence. Examples of new social or community-based information features include enhancing the quality of traditional Web search and information retrieval approaches by leveraging the inherent social connections between users and other users via social networks, social tagging, and other community-based features (e.g., social collective intelligence) [4,9,24]. Such systems are noted for their open and unregulated nature, self-supervision, and dynamism, which are key features in supporting information sharing and knowledge discovery.

But these features also place individuals and their computer systems at risk for abuse and exploitation at the hands of malicious adversaries who seek to exploit the perceived social bonds inherent in social information systems. Some observed vulnerabilities include impersonated (or fraudulent) digital identities [35], targeted malware dissemination [7], social network enhanced phishing [19], and corrupt user-generated metadata (or tags) [22]. Understanding different types of social spam and deception is the first step towards countering these vulnerabilities. The second and more challenging step is to provide dependable capabilities for knowing whom to trust and what information to trust, given the open and unregulated nature of these systems.

With these problems in mind, we focus on building an online community platform that allows wide access to many different types of users and that still remains useful, even in the presence of users intent on manipulating the system. As a first step towards this goal, we propose SocialTrust, a reputation-based trust aggregation framework for supporting tamper-resilient trust establishment in online communities. The benefits of reputation-based trust from a user's perspective include the ability to rate neighbors, a mechanism to reach out to the rest of the community, and some assurances on the trustworthiness of unknown users in the community. Reputation systems are an important feature of many e-marketplaces and online communities (like eBay, Amazon, and Digg), and reputation-based trust systems have received considerable attention in P2P systems (e.g., [1,21,26]). Most existing approaches, however, ignore the social constructs and social network topology inherent in online communities, and typically provide less personalized criterion for providing feedback and computing reputations.

A key challenge then is whether we can develop a trust model for online communities that is tamper-resilient even in the presence of malicious users. And what are the critical factors impacting such a model? We believe that understanding the dynamics of trust establishment can have wide-ranging impact in large-scale collaborative digital libraries, in question answering communities like Yahoo! Answers, in Wikipedia-style information sharing communities, and in other community-based information management systems.

In particular, we carefully consider the unique properties of social networks to build the SocialTrust model for tamper-resilient trust establishment in online communities. Three of the salient features of SocialTrust are:

- *Incorporating personalized user feedback* – SocialTrust augments the relationships in the social network with a personalized feedback mechanism so that a user's trust rating can reflect his behavior (via feedback) as well as the user's position in the social network.
- *Distinguishing user relationship quality from trust* – Many trust approaches make no distinction between the trust placed in a user and the trust placed in a user's relationships. SocialTrust incorporates these distinct features, leading to better resistance to trust manipulation.
- *Tracking user behavior* – SocialTrust incorporates the evolution and trajectory of a user's trust rating to incent long-term good behavior and to penalize users who build up a good trust rating and suddenly "defect."

We experimentally evaluate the SocialTrust framework over a simulated information sharing community based on real social network data consisting of millions of MySpace profiles and relationships. While other trust aggregation approaches have been developed and implemented by others, we note that it is rare to find such a large-scale experimental evaluation. We find that in the context of large-scale attempts to undermine the quality of ratings that it is significantly more robust than popular alternative trust models.

2. Background

In this section, we identify several vulnerabilities to the quality of information in a social information sharing community, describe the reference model for an online community, and briefly describe some baseline trust measures.

2.1. Vulnerabilities in online social networks

While there are important problems associated with securing the social network infrastructure (e.g., ensuring that profiles are correctly formatted and contain no browser exploits, containing denial-of-service attacks), we explore vulnerabilities to the quality of information available through online social networks even when the underlying social network infrastructure has been secured.

In particular, we identify three important vulnerabilities:

- *Malicious infiltration:* Unlike the Web-at-large, most online social networks do provide some limits as to who can and cannot participate. Users typically must register with an online social network using a valid email address or by filling out a

user registration form. As a result, many social networks give the illusion of security [5], but malicious participants can gain access as has been observed in MySpace [35] and the more closely guarded Facebook [10].

- *Nearby threats:* Most online social networks typically enforce bilateral agreement between both parties in a relationship. As a result, participants in a social network have tight control over who their friends are. There is a well-noted illusion of privacy in online social networks [8,30] in which participants have a lack of understanding of the potential threat of participants two or more hops away. The small world phenomenon – a feature of many social networks [28,38,39] – means that there is a short distance in the network between any two participants. So even if a user has tight control over his direct friends, malicious participants can be just a few hops away from any participant. For example, we have identified a deceptive rogue profile on MySpace with nearly 200 declared friendships with (seemingly) legitimate community members, indicating that some community members are either easily deceived or have low standards for friendship.
- *Limited network view:* Even if a user in the social network maintains tight control over her friends *and* closely monitors the quality of her neighbors' friends, she will still have access to only a limited view of the entire social network. Since any one user may have direct experience or relationships with only a small fraction of all social network members, she will have no assurances over the vast majority of all participants in the network.

Malicious users can exploit the perceived social connection between users for increasing the probability of disseminating misinformation, of driving participants to the seedy side of the Internet (e.g., to sites hosting malware), and of other disruptions to the quality of community-based knowledge.

To counter these vulnerabilities, we can rely on a number of approaches including legal enforcement, background checks, and reliance on a trusted central authority. A strictly legal approach is one adopted by many of the popular social networking sites, in which participants who are deemed to be in violation of the terms of service may be identified and removed from the social network. Complementary to this approach, a trusted central authority may be empowered to monitor the community (as in MySpace and Facebook). Alternatively, users may be required to undergo a background check to provide some off-line assurances as to their quality. These approaches typically suffer from problems with enforcement and scalability. In this manuscript, we use a *reputation-based trust approach* for maintaining the relative openness of these communities (and the corresponding benefits) and still providing some measure of resilience to the described vulnerabilities.

2.2. Reference model

The reputation-based trust model represents an online social network $\mathscr{SN}$ as a triple consisting of profiles $\mathscr{P}$, relationships $\mathscr{R}$, and contexts $\mathscr{C}$: $\mathscr{SN} = \langle \mathscr{P}, \mathscr{R}, \mathscr{C} \rangle$.

The most basic element in a social network is a *profile*. We most typically think of profiles in terms of specific people (e.g., me, my friends, my parents, etc.), but profiles may represent companies, products, musical groups, abstract concepts (like love, harmony, and peace), and so on. Typically, a profile is a user-controlled Web page that includes some descriptive information about the person it represents. Most profiles include some personal information like the person's picture, age, gender, location, and interests. Additionally, a profile may be augmented with a collection of digital artifacts like Web documents, media files, etc. For simplicity in presentation, we will refer to profiles and users interchangeably in the rest of this manuscript. We denote the set of all profiles in the social network $\mathscr{SN}$ as $\mathscr{P}$. We shall assume there are n profiles in the network, numbered from 1 to n: $\mathscr{P} = \{p_1, \ldots, p_n\}$.

We denote a relationship between profiles i and j with two entries in the relationship set $\mathscr{R}$ to characterize each participant's contextual view of the relationship: $rel(i, j, c_1)$ and $rel(j, i, c_2)$, where c_1 and c_2 are two contexts drawn from the context set $\mathscr{C}$. We denote user i's set of contacts as $rel(i)$ and the total number of relationships i participates in as $|rel(i)|$. A relationship in a social network is a bidirectional link between two users. A relationship is only established after both parties acknowledge the relationship. The context indicates the nature of the relationship – e.g., the two people are co-workers.

2.3. Baseline trust measures

Given the reference model, we next identify two baseline trust measures that serve as a point of contrast for the SocialTrust approach developed in the following section. The first measure is a simple popularity-based measure that considers the sheer quantity of relationships that a user engages in for evaluating the trustworthiness of participants:

$$PopTrust(i) = |rel(i)| \tag{1}$$

This relationship count has close analogs in other network analysis domains, including bibliometrics and traditional social network analysis (where popularity can be measured by a count of contacts or friends) [29].

A more sophisticated trust measure is a PageRank-style random walk trust measure, in which a random walker surfs the social network much like the random surfer in the popular PageRank approach for Web ranking [32]. Viewing the social network $\mathscr{SN} = \langle \mathscr{P}, \mathscr{R}, \mathscr{C} \rangle$ as a graph where the profiles $\mathscr{P}$ are nodes and the relationships $\mathscr{R}$ are labeled directed edges, the random walker proceeds across the graph; at each node i, the random walker follows one of i's relationship links with probability $1/|rel(j)|$. Occasionally the random walker resets to a new user with a reset probability of $1/n$ for all i.

$$RWTrust(i) = \lambda \sum_{j \in rel(i)} RWTrust(j)/|rel(j)| + (1 - \lambda)\frac{1}{n} \tag{2}$$

In the long run, the random walker will visit high-quality users more often than low-quality ones.[1] Such random walk models have been studied in both the peer-to-peer file-sharing domain (EigenTrust) [21] and in the context of trust management for the Semantic Web [33]. The TrustRank approach for Web ranking extends the basic model by exchanging the random reset probability with some a priori notion of pre-trusted users [18]. More recently, similar random walk models have been applied to social networks (where nodes are users and links are the relationships between users) in [42] and studied more closely in the context of expertise networks in [44]. In all of these models, a link (or relationship) from one node to another is a crude quality indicator of the target node. While a single link may not confer great confidence in the quality (or trust) of a user, in the aggregate, we might expect the many links to a user (and recursively, the many links to user's who link to a target user) to provide a signal about a target user's trustworthiness. However, as we will see in our experimental study, a straightforward application of these random walk models to an information sharing community ignores some of the key features of social networks and results in poor information quality (since malicious users are being treated like trusted users).

3. The socialtrust framework

Given the preliminary trust models, we now turn our attention to building the SocialTrust framework. Three key observations motivate our current work:

1. *Incorporating personalized user feedback.* Trust models like PopTrust and RWTrust are based solely on network topology and are divorced from the underlying behavior of the users in the network. Relationships in the online social network provide the basis for trust aggregation, but there is no feedback mechanism for dynamically updating the quality of the trust assessments based on how well each user in the network behaves. Hence, we are interested in "closing the loop" so that the trust assessments may be dynamically updated as the social network evolves and as the quality of each user (with respect to user feedback) changes over time.
2. *Distinguishing user relationship quality from trust.* Second, many trust models (e.g., [18,21]) evaluate the relative trustworthiness of a node (or user, in our case) based on the trustworthiness of all nodes pointing to it, but make no distinction about the relationship (or link) quality of each node. In essence, these approaches make no distinction between the trust placed in a user and the trust placed in a user's relationships. Intuitively, we would like to differentiate between users who consistently engage in high-quality relationships with other users versus users who tend to engage in lower quality relationships.
3. *Tracking user behavior.* Third, we are interested in robust trust measures that can incorporate the evolution and trajectory of a user's behavior in the network. Trust models like PopTrust and RWTrust provide a snapshot of the trustworthiness of a user based on the current state of the social network, but do not consider the history or the immediate change in behavior of the user. A trust rating should incorporate features to incent long-term good behavior and to penalize users who build up a good trust rating and suddenly "defect."

Based on these observations, we next introduce the overall SocialTrust framework. The goal of SocialTrust is to enhance online communities by providing a trust rating for each user by leveraging the rich social connections of the social network. SocialTrust is explicitly designed to (i) leverage the relationships inherent in the social network; (ii) gracefully handle the addition of new users to the network as it evolves; and (iii) be robust against efforts to manipulate the trust ratings.

3.1. Assessing trust with socialtrust

We denote the SocialTrust trust rating of user i at time t by $ST(i,t)$. For any two users in the community, we may evaluate the relative trustworthiness, e.g., that user i is more trustworthy than user j (i.e., $ST(i,t) > ST(j,t)$). This aggregated trust information may be used by users for enhancing the quality of their experiences in the community. Since users will typically have direct relationships with only a small fraction of all users in the network, trust values may be used to evaluate the quality of the vast majority of other users for which the user has no direct experience.

For presentation clarity, we shall assume the presence of a centralized *trust manager* whose job is to compute trust ratings for users in the network and to communicate these trust ratings to users when needed. Alternatively, the duties of the trust manager may be securely distributed throughout the network (see, for example, [20]).

Initially all users are treated equally. SocialTrust supports trust maintenance through dynamic revision of trust ratings according to three critical components: the current quality component of trust $Tr_q(i,t)$, the history component, and the adaptation to change component.

$$ST(i,t) = \alpha \cdot Tr_q(i,t) + \beta \cdot \frac{1}{t}\int_0^t I(x)Tr_q(i,x)dx + \gamma \cdot Tr'_q(i,t) \tag{3}$$

[1] For presentation clarity we are assuming that a user's relationship edges are treated equally. It is fairly straightforward to generalize to the case of arbitrary strength; in terms of the random walk, relationship edges will be followed with non-uniform probability according to the relationship strength.

where $Tr'_q(i,t)$ is the derivative of $Tr_q(i,x)$ at $x = t$, where $I(x)$ is an importance function, and where α, β, and γ are tunable parameters. This approach is similar to a Proportional-Integral-Derivative (PID) controller used in feedback control systems [31].

3.2. The quality component of trust

The first component of SocialTrust is the quality component $Tr_q(i,t)$ which provides the system's view of the trustworthiness of the user based on the current state of the social network. For simplicity in presentation, we shall drop the interval notation when the context is clear (e.g., $Tr_q(i)$ vs. $Tr_q(i,t)$). Following the observations at the beginning of this section, the quality component incorporates a feedback rating $F(i)$ and a user's relationship quality score $R(i)$ for augmenting the random walk models presented in the previous section:

- *Trust group feedback:* For any user in the network, we establish a trust group that governs which other participants in the social network a user can make an assessment of (and which other participants can make an assessment of that user). For each trust group, we aggregate the feedback rating of participants in the network, giving each user i a feedback rating $F(i)$.
- *User's relationship quality:* Secondly, we measure each user's relationship quality, denoted $R(i)$. Relationship quality measures the quality of a user's relationships in the social network. One of the benefits of user relationship quality is that it provides an incentive for users to monitor the quality of their relationships.

With personalized feedback and user relationship quality, we propose the core SocialTrust trust metric. The intuition is that a user's trustworthiness should be determined by: (i) the number and trustworthiness of the users who are in a relationship with her; (ii) the relationship quality of each of these users; and (iii) the feedback rating of each user. In this way, a relationship edge from a high-trust/high-relationship-quality user counts more than a relationship edge from a high-trust/low-relationship-quality user. By decoupling relationship quality from the user's trustworthiness, we can determine the relationship-quality-augmented trust of each user.

$$Tr_q(i) = \lambda \sum_{j \in rel(i)} R(j) \cdot Tr_q(j)/|rel(j)| + (1-\lambda)F(i) \tag{4}$$

Compared to the baseline random walk trust model (recall Eq. (2)), this formula states that the trustworthiness of user i is determined by the trustworthiness $Tr_q(j)$ and the relationship quality $R(j)$ of the users that engage in a relationship with her, as well as by the number of user j's relationships (via the factor $|rel(j)|$).[2] In this sense, the relationship edge weights are used to determine how a user's "vote" is split among the users, but the relationship quality of a user impacts how large or small is the user's vote. The feedback rating $F(i)$ favors users who have been rated highly by other users within the trust establishment scope, according to the mixing factor $1-\lambda$. In our evaluation, we shall validate that this approach does, in fact, lead to more tamper-resilient trust ratings than popular alternative trust models.

3.3. History and adaptation to change components of trust

The trust quality component $Tr_q(i,t)$ indicates how well the system believes that user i can be trusted at a point-in-time, but without any consideration of user i's behavior in the past nor any consideration for sudden changes in a user's behavior. Hence, the second and third components of SocialTrust (recall Eq. (3)) consider the evolution and trajectory of a user's trust rating.

The history component – $\frac{1}{t}\int_0^t I(x)Tr_q(i,x)dx$ – considers the integral of the trust value over the lifetime of the user in the network, say, from time 0 to the current time t, weighted by an importance function I. This history component is important for (i) providing an incentive to all users in the network to behave well over time; and (ii) limiting the ability of malicious participants to whitewash their trust ratings by repeatedly leaving and re-entering the network. The importance function I allows us to optimize the history component by balancing the weight given to more recent periods versus less recent periods.

The adaptation to change component – $Tr'_q(i,t)$ – tracks shifts in a user's behavior. This change component can mitigate the impact of malicious participants who build up a good trust rating over time (through the other two components) and suddenly "defect."

4. Developing and refining socialtrust: feedback, user relationship quality, and tracking user behavior

In the previous section, we saw how the overall SocialTrust approach augments and extends the baseline random walk trust model on at least three counts. First, the core SocialTrust metric $Tr_q(i,t)$ supports personalized feedback through personalized trust group formation. Second, it distinguishes user relationship quality from trust. Third, it incorporates trust history

[2] Contextual information (recall the context set $\mathscr{C}$) can be used to revise this uniform split, for example, to favor relationships with friends and family over relationships with co-workers.

and change adaptation, which are critical factors to ensure quality trust ratings over time. In the rest of the manuscript, we focus our attention on these three features to understand how they impact the quality of the overall SocialTrust approach:

- Section 4.1: Assessing personalized trust group feedback
- Section 4.2: Distinguishing user relationship quality from trust
- Section 4.3: Tracking user behavior

In the following sections, we address each of these features in turn to provide a thorough understanding of SocialTrust and how it supports robust trust establishment. We provide some guidance on how these questions could be resolved, provide an informal analysis of the vulnerabilities that each factor tackles, and validate these choices on a large-scale experimental study over millions of real social networking profiles.

4.1. Assessing personalized trust group feedback

The first feature we study is the feedback rating $F(i)$. The SocialTrust quality component (recall Equation (4)) incorporates this user-driven feedback of other users to augment the trust assessment of each user with user feedback, so that the user's relative place in the social network – e.g., very popular in terms of relationships – is not the sole arbiter of trustworthiness. By comparison, the alternative trust models PopTrust and RWTrust (recall Equations (1) and (2)) are based exclusively on the relationship structure of the social network topology and hence are divorced from the underlying behavior of the users in the network.

For example, PopTrust is subject to extreme manipulation by malicious (or even just ego-centric) participants. Since online identities are cheap (often requiring only a valid email address for authentication), malicious cliques can form in which many new users join the network for subverting the trust model; each user in the clique maintains a relationship with a specific target user, resulting in an arbitrarily high popularity and, hence, trust rating for the target user. Similar brute-force attacks are possible on RWTrust as well, and these types of attacks have been well-studied in the Web domain (e.g., [17]). A sufficiently large malicious clique can boost internal users by mutual recommendation promotion. These types of attacks exploit the purely topological aspect of these trust measures.

Hence, we are interested in "closing the loop" so that the trust assessments may be dynamically updated as the social network evolves and as the quality of each user (with respect to user feedback) changes over time. In this manuscript, we examine user behavior with respect to information sharing, but this general principle may be applied to other types of behaviors in social networks. In the following sections, we build our proposed trust group feedback mechanism and then experimentally validate it versus these baseline trust measures in our experiments section.

4.1.1. Trust establishment scope

For any user in the network, we establish a trust group that governs which other participants in the social network a user can make an assessment of (and which other participants can make an assessment of that user). For each trust group, we aggregate the feedback rating of participants in the network, giving each user i a feedback rating $F(i)$.

The trust establishment scope governs what other participants in the network each user can judge, and what other participants can judge each user. Trust group formation can be tuned to balance efficiency and the security of the overall system (by constraining users from manipulating the reputation of users outside of their trust group). At one extreme, there is a single *trust group* consisting of all members of the social network. At the other extreme, each user belongs to a lone trust group consisting of only themselves, meaning that the system supports no trust aggregation. For balancing these two extremes, we could rely on trust groups defined by self-described interests (e.g., sports), location (e.g., members who all live in Texas), or other contextual information.

In this manuscript, we propose to define trust groups based on the chains of relationships that are fundamental to the formation of social networks. Hence, we consider a *relationship-based* model for determining a user's trust group where the size of the trust group is determined by a network-specified *radius*, ranging from a user's direct neighbors (radius 1), to a user's direct neighbors plus his neighbors' neighbors (radius 2), and so on. By limiting the radius of a user's trust group, we can constrain the impact of malicious users who are far away in the social network. In practice, we form relationship-based trust groups through the *browse-based search capability* provided by most online social networks, whereby a user's profile may be viewed (or browsed) by other users. Users may manually browse from profile to profile and provide ratings to the trust manager on users encountered subject to the radius of the relationship-based trust group.

Given a trust group, we next describe several strategies for assessing the trust group feedback in SocialTrust. We assume that each user i in the network is associated with a feedback value $F(i)$ that indicates how well the user's trust group views the user. The feedback ratings are taken from the interval $[0, 1]$. We make two observations: (i) user behavior is dynamic, so the feedback ratings should be dynamically updated; and (ii) malicious users may attempt to subvert them.

For assessing feedback ratings, each user maintains state about the other users it has made a rating for through browse-based search. Based on the ratings of all users who have interacted with user j, we can assess a feedback rating $F(j)$. Guaranteeing that feedback ratings are robust to manipulation is an important feature, and there have been several recent studies on how to ensure such robustness (e.g., [34,36,41]) in addition to securing the voting infrastructure (e.g., through encrypted votes, secure transmission, etc.).

We briefly describe five important factors (as identified in [41]) that any feedback mechanism should consider (where we adapt these factors to online social networks):

- *Satisfaction:* When user i encounters user j either through the browsing process or through a query response, how well is user i satisfied with user j? This satisfaction level is the basis for assessing user j's overall feedback rating.
- *Number of interactions:* For how many interactions does a user satisfy the originating user? For example, if user i satisfies 10 queries in the most recent period, but user j satisfies 999 out of 1000, which user should be deemed of higher quality? A clique of malicious participants may engage in fake interactions with each other to mask their poor behavior to legitimate users in the network.
- *Feedback credibility:* How credible are the users providing the feedback? Some users may habitually provide non-truthful feedback ratings while others are truthful. It is important to understand each user's credibility to prevent gaming of the feedback system through dishonest feedback.
- *Interaction context factor:* Some interactions between users may be more important than others, and the interaction context factor is intended to capture this relative value. For example, a user who provides high-quality job lead information for a high-ranking executive could be rewarded more in terms of feedback rating than a user who provides less valuable information.
- *Community context factor:* Finally, the online community itself may possess community-specific properties that impact the feedback mechanism – e.g., rewarding members who provide feedback as an incentive for encouraging participation in the feedback process.

Based on these observations, we develop a feedback aggregation approach. We rely on a fairly basic rating scheme to show the power of the SocialTrust framework even without these more sophisticated techniques; we anticipate revisiting this issue in future work. A vote is a pair of the form $\langle user, vote \rangle$, where *user* is a unique user identifier (the profile number) and *vote* is either "good" or "bad". Each user communicates to the trust manager a vote for user it has interacted with in the most recent period. We consider three voting schemes – (i) open voting; (ii) restricted voting; and (iii) trust-aware restricted voting. We describe the first two and their drawbacks to motivate the final trust-aware restricted voting scheme.

Open voting: We use the shorthand $v_i(j)^+$ to indicate a "good" vote by user i for user j; $v_i(j)^-$ indicates a "bad" vote. In the simplest case user j's feedback rating $F(j)$ is the fraction of "good" votes cast for user j:

$$F(j) = \frac{\sum_i \mathcal{I}(v_i(j)^+)}{\sum_i \mathcal{I}(v_i(j)^+) + \mathcal{I}(v_i(j)^-)}$$

where the indicator function $\mathcal{I}(\cdot)$ resolves to 1 if the argument to the function is true, and 0 otherwise. This open voting policy is subject to ballot stuffing. A single malicious user can issue an unlimited number of "good" votes for raising the feedback rating of colluding users or can issue "bad" votes for demoting the feedback rating of competing users.

Restricted voting: We can restrict how much each user can vote by assigning each user a limited number of *points* to be allocated over all of its votes. We let w_{ij} denote the number of points user i uses to weight her vote for user j, where the total points allocated to each user is an arbitrary constant: $\sum_j w_{ij} = 1$. Hence, this restricted voting leads to a new feedback rating:

$$F(j) = \frac{\sum_i w_{ij} \mathcal{I}(v_i(j)^+)}{\sum_i w_{ij} \mathcal{I}(v_i(j)^+) + w_{ij} \mathcal{I}(v_i(j)^-)}$$

The trust manager will only accept up to $\sum_j w_{ij} = 1$ points per voter i. All votes over the restriction will be ignored. By restricting the total size of vote allocated to each user, this restricted voting scheme avoids the problem of vote stuffing by a single user. We have no assurances that a malicious user will choose to vote truthfully for other users it has actually interacted with, but we do know that the total amount of voter fraud is constrained. Unfortunately, such a voting scheme is subject to collusive vote stuffing, in which many malicious users collectively decide to boost or demote the feedback rating of a selected user.

Trust-aware restricted voting: To handle the problem of collusive vote stuffing, we advocate a weighted voting scheme in which users are allocated voting points based on how trustworthy they are. We again let w_{ij} denote the number of points user i uses to weight her vote for user j, but now the total points allocated to each user depends on her trustworthiness: $\sum_j w_{ij} = ST(i)$. This trust-aware restricted voting scheme results in a feedback rating for user j of:

$$F(j) = \frac{\sum_i ST(i) w_{ij} \mathcal{I}(v_i(j)^+)}{\sum_i ST(i) w_{ij} \mathcal{I}(v_i(j)^+) + ST(i) w_{ij} \mathcal{I}(v_i(j)^-)}$$

The trust manager will only accept up to $\sum_j w_{ij} = ST(i)$ points per voter i. All votes over the restriction will be ignored, meaning a malicious user cannot ballot stuff. If a malicious user receives poor feedback from trusted users in the system, then his feedback rating will be negatively affected, which in turn will impact his trustworthiness in the system. Intuitively, this cycle is appealing since it can dynamically adapt to trusted users who over time begin behaving badly as well. Note that other feedback approaches are possible and easily pluggable into the SocialTrust framework.

4.2. Distinguishing user relationship quality from trust

Given the personalized feedback of SocialTrust, we next turn to the second of the three key features in our development of the trust framework: distinguishing user relationship quality from trust. Recall that the SocialTrust quality component Eq. (4) assesses the trust score of user i by recursively considering the trust score $Tr_q(j)$ and the relationship quality $R(j)$ of users in a relationship with user i. Compared to the baseline RWTrust model (recall Equation (2)), the relationship quality scores are used to modulate how large or small is a user's trust contribution to her neighbors. In essence, the baseline random walk models (e.g., [18,21]) make no distinction between the trust placed in a user and the trust placed in a user's relationships. Intuitively, we would like to differentiate between users who consistently engage in high-quality relationships with other users versus users who tend to engage in lower quality relationships.

Recall that user i participates in a total number of relationships $|rel(i)|$. How many of these relationships are with high quality users? Our goal in this section is to formally assess the quality of a user's relationships. Concretely, let $R(i)$ denote the *user relationship quality* of user i. A score of $R(i) = 0$ indicates that user i has poor quality relationships. In contrast, a score of $R(i) = 1$ indicates that user i has high quality relationships.

The small world nature of many social networks means that a large portion of the network may be reachable from any one user within a few hops. Hence, a user's relationship quality should depend on the user's direct relationships and perhaps the relationships of its neighbors up to some small number (k) of hops away. We also observe that a user's relationship quality should be related to the feedback ratings of its neighbors. A user who only engages in relationships with well-behaving users should earn a higher relationship quality score than a user who has relationships with poorly behaving members of the network. We next formally define user relationship quality and provide a discussion of the factors impacting its assessment.

4.2.1. User relationship quality as a scoped random walk

We model the relationship quality of user i in terms of a scoped random walk model, in which a random walker originates its walk at user i and randomly follows the relationship edges of user i and the subsequent users at which it arrives up to some small number of steps. The random walker will choose to terminate his random walk after walking up to small number of steps away from the originating user. The scoped random walk considers the feedback rating of each user to help guide his walk.

In the extreme, when all users within k hops of the original user i have a perfect feedback rating (i.e., $F(j) = 1$ for all users within k hops of i), then user i has relationship quality $R_k(i) = 1$. In contrast, if user i either has a poor feedback rating (i.e., $F(i) = 0$) or all of the users within k hops of user i have poor feedback ratings, then user i's relationship quality is $R_k(i) = 0$. To summarize, the relationship quality of user i can be interpreted as the probability that a random walker originating its walk at user i ends at a high-quality user after walking up to k-hops away from i.

We can begin our examination of user relationship quality by considering the base case when the scope (k) is 0.

Base case (k=0): In the base case, the relationship quality of a user is merely its feedback rating $F(i)$:

$$R_{[0]}(i) = F(i)$$

The random walker walks for 0-hops, meaning that it stays at the original user. The probability that the random walker ends at a high-quality user is thus $F(i)$. For users new to the network or for users lacking any feedback, a default feedback rating can be assigned for guiding the relationship quality assessment.

One-hop case (k=1): In the one-hop case, the relationship quality of a user is the probability that the random walker ends at a high-quality user after walking forward to one of the contacts from the original user's set of relationships (recall that $rel(i)$ denotes the relationship list for user i):

$$R_{[1]}(i) = F(i) \sum_{j \in rel(i)} F(j)/|rel(i)|$$

Note that the random walker proceeds initially according to the feedback rating $F(i)$ of the original user. Accordingly, the relationship quality of a user that has received poor feedback will be low. But a user with a high feedback rating who engages in relationships with poor quality users will also be penalized with low relationship quality.

Two-hop case (k=2): The user relationship quality can be extended to consider random walks of length two, where:

$$R_{[2]}(i) = F(i) \sum_{j \in rel(i)} F(j)/|rel(i)| \left[\sum_{l \in rel(j)} F(l)/|rel(j)| \right]$$

We can extend relationship quality to consider random walks of arbitrary length k. In all cases, user relationship quality is a local computation and can be updated in a straightforward fashion requiring only an originating user and a forward crawl of all users within k hops of the originating user. Hence, the relationship quality of a user can be updated in a straightforward fashion, without the need for an expensive computation.

4.2.2. Correction factor

The scoped random walk provides a natural measure of the relationship quality of each user. However, the feedback ratings used for driving the user relationship quality assessment may not be known with certainty and malicious users may attempt to subvert these ratings (recall Section 4.1). Hence, in this section, we discuss several correction factors for augment-

ing the basic scoped random walk model in the presence of such uncertainty. We denote the updated user relationship quality score for user i as $\widehat{R}_{[k]}(i)$, and evaluate it in terms of the original user relationship quality score and a correction factor ϕ:

$$\widehat{R}_{[k]}(i) = \phi \cdot R_{[k]}(i)$$

We present an optimistic and a pessimistic correction factor as two baseline approaches to motivate a hop-based correction factor. The hop-based factor balances the extremes of and pessimistic factors for guiding the proper relationship quality correction factor for each user.

Optimistic correction: The optimistic correction factor makes no changes to the original user relationship quality as determined by the scoped random walk. For all users, the optimistic correction factor is 1:

$$\phi_{opt}(i) = 1, \forall i$$

The optimistic approach will tend to over-estimate the relationship quality of users that (i) are part of a malicious clique in which some users behave well to mask their relationships with clique members who behave poorly; or (ii) engage in relationships with poor quality users for whom the feedback ratings have incorrectly identified as high quality.

Pessimistic correction: The pessimistic correction factor treats a user with even a very small likelihood (call it δ) of engaging in a relationship with a poorly performing user as if all of the user's relationships were with users of low feedback rating.

$$\phi_{pess}(i) = \begin{cases} 0 & \text{if } R_{[k]}(i) < 1 - \delta \\ 1 & \text{otherwise} \end{cases}$$

A pessimistic approach may be appropriate in circumstances when relationships with malicious users are highly correlated (as in a malicious clique) or when malicious users in the network are considered extremely dangerous. In this second case, even a single relationship with such a dangerous user would warrant a severe correction to the user's relationship quality.

Hop-based correction: In contrast, the hop-based correction factor seeks to provide a balance between the optimistic and pessimistic correction factors by considering the number and the length of the paths emanating from a user that reach bad users. A *path* in the social network from user i to user j is a sequence of users: $path(i,j) = \langle x_0, x_1, \ldots, x_n \rangle$ (where $i = x_0$ and $q = x_n$) such that there exists a directed relationship edge between successive nodes in the path, $x_{l+1} \in rel(l)$, for $0 \leqslant l \leqslant n-1$. We say a path reaches a bad user if the feedback rating for the user is less than some threshold δ. We call such a path a *bad path*.

For a bad path of length l originating at user i, we associate a hop-based correction factor $\phi_{hop,l}(i)$, where $0 \leqslant \phi_{hop,l}(i) \leqslant 1$. By default, we let $\phi_{hop,l}(i) = 1$ if there are no bad paths of length l originating from i. The hop-based discount factor can then be calculated as the product of the constituent discount factors: $\phi_{hop}(i) = \prod_{l=1}^{k} \phi_{hop,l}(i)$.

Selecting the appropriate hop-based correction factor is important, and there are a number of possible approaches. In this manuscript, we advocate an exponentially decaying correction factor. Beginning with a user-defined factor $\psi(0 < \psi < 1)$ to set the initial hop-based correction for bad paths of length 1, i.e., $\phi_{hop,1}(i) = \psi$, the exponential approach tunes this correction closer to 1 as the bad path length increases:

$$\phi_{hop,l}(i) = 1 - (1 - \psi)\psi^{l-1}$$

meaning that longer bad paths result in a less severe correction to a user's relationship quality than do shorter paths. Starting with an initial correction factor ψ close to 0 will result in a more pessimistic correction, whereas ψ close to 1 is intuitively more optimistic.

4.3. Tracking user behavior

Finally, we turn our attention to the third of the three key features in our development of the trust framework: tracking user behavior. Recall that the overall SocialTrust framework Eq. (3) assigns a trust score to user i based on the current state of the network (via the trust quality component $Tr_q(i,t)$, of which feedback and user relationship quality are key features) and the history and immediate change in behavior of the user via the integral (history) and derivative (change) components. The history and change components of SocialTrust are important to track the evolution and trajectory of a user's trust rating for a more robust overall trust score.

Compared to the overall SocialTrust framework Eq. (3), the alternative baseline trust models – PopTrust and RWTrust – provide only a snapshot of the trustworthiness of a user based on the current state of the social network. A snapshot trust measure alone does not provide proper incentive for a user to engage in long-term behavior nor does it limit the ability of malicious participants from repeatedly leaving and re-entering the network to whitewash their trust ratings. In both of these cases, it would be reasonable to consider the history of a user's behavior as an input to the user's current trust rating. However, even with the history of a user's behavior, a malicious user may suddenly "defect" and begin acting improperly within the particular application scenario (as in information sharing as we study in this manuscript). Hence, we can also add in a change component to the trust rating to penalize sudden shifts in behavior.

4.3.1. Implementing the history and change components of trust

In practice, the SocialTrust trust computation will be launched at periodic intervals or on a per-need basis. Assuming that the trust computation is launched at regular intervals for some maximum history H, then the trust manager has access to M

historical trust values for user i, denoted $ST(i,m)$ for $1 \leqslant m \leqslant M$, where interval M is the most recent interval. Based on these historical trust values, we can calculate the discretized version of the historical and change adaptation components of trust. For simplicity in presentation, we shall drop the interval notation when referring to the most recent interval (e.g., $ST(i)$, $Tr_h(i)$).

Historical SocialTrust ratings [$Tr_h(i)$]: We can evaluate the history component of trust, denoted $Tr_h(i)$, as a weighted sum over the M historical trust values:

$$Tr_h(i) = \sum_{m=1}^{M} ST(i,m) \cdot \frac{I_m}{\sum_{l=1}^{M} I_l}$$

where I_m is an importance weight associated with the trust value computed during the m'th time interval. One choice of importance weight is the exponentially weighted sum: $I_m = \xi^{M-m}$ for $1 \leqslant m \leqslant M$. For $\xi < 1$, the time-averaged trust favors more recent scores for a user over older trust scores. Letting $\xi = 1$ results in a simple average of the trust scores for the history.

Adapting to change in behavior [$Tr_c(i)$]: The change adaptation component of trust, denoted $Tr_c(i)$, considers the current SocialTrust rating for a user relative to the user's past behavior (where the entire history component is used to reflect previous behavior for stability reasons):

$$Tr_c(i) = ST(i) - Tr_h(i)$$

Alternate approaches for the change adaptation component could consider trust changes over the previous N ($N < M$) periods instead of the entire history to emphasize deviations from the user's short-term trust rating.

Putting it all together: Finally, as compared to the original continuous version of SocialTrust (see Eq. (3)), the overall discretized SocialTrust rating for a user i for the most recent interval can be calculated as:

$$ST(i) = \alpha \cdot Tr_q(i) + \beta \cdot Tr_h(i) + \gamma(Tr_c(i)) \cdot Tr_c(i) \tag{5}$$

where $\gamma(x) = \gamma_1$ if $x \geqslant 0$ and $\gamma(x) = \gamma_2$ if $x < 0$. The γ_1 and γ_2 weights impact the overall amplification factor of the change adaptation component of trust. In practice, we would expect to associate a higher weight to γ_2 to more severely degrade a user's trust rating if he engages in sudden poor behavior.

4.4. Tracking user behavior in action

As illustrated in the overall SocialTrust framework Eq. (3) and the discretized version Eq. (5), there are three tunable knobs to balance the current quality component of trust (α), the history component (β), and the change component (γ). By tuning α, β, and γ, the SocialTrust model can be optimized along a number of dimensions, e.g., (i) to emphasize the most recent behavior of a user in the network (by choosing higher values of α); (ii) to de-emphasize the current user's behavior in the context of his entire history of behavior (by choosing higher values of β); or (iii) to amplify sudden fluctuations in behavior (by choosing higher values of γ). In addition, the history and change adaptation components of trust allow the overall SocialTrust rating to tolerate errors in the calculation of the node's current trust rating ($Tr_q(i,t)$).

To illustrate the impact of the history and change components on the overall SocialTrust framework, we consider in Figs. 1 and 2 two scenarios. In both scenarios, we consider a baseline user who behaves poorly for 10 time steps (with trust rating 0), then behaves well for 10 time steps (with trust rating 1.0), then back down again to 0 for 10 time steps. This corresponds to a case where a user strategically alternates its behavior in an effort to disrupt the application scenario in which the trust measurement is deployed (e.g., information sharing, as in this manuscript). In both scenarios, the baseline case is the snapshot trust rating for the user based solely on the quality component of trust $Tr_q(i)$.

In the first scenario (Fig. 1), we compare the baseline case to two settings of the SocialTrust framework that include the quality component of trust and the history component of trust. One approach favors the current behavior over past behavior ($\alpha = 0.8;\ \beta = 0.2$). The other approach favors the past behavior over current behavior ($\alpha = 0.2;\ \beta = 0.8$). In both cases, we consider a history of $M = 5$ time steps and an importance weight $\xi = 0.7$. With $\alpha = 0.2$, the SocialTrust trust score takes several steps to build from 0 to 1, which is a desirable property of a trust metric in which we expect the user to slowly build trust. The downside of such an approach however is illustrated when the user's behavior drop from 1 to 0. In this case, the more-recent leaning SocialTrust approach with $\alpha = 0.8$ drops much faster, which means that user's have less ability to take advantage of the application setting (by appearing to be a high-trusted user, even though the behavior has dropped to 0).

In the second scenario (Fig. 2), we again capture the baseline snapshot trust rating for the user based solely on the quality component of trust $Tr_q(i)$. Alternatively, we also consider the approach considering only the quality component of trust and the history component of trust as in the previous scenario – with no adaptation to change component ($\alpha = 0.2;\ \beta = 0.8;\ \gamma_1 = 0;\ \gamma_2 = 0$). The other approach includes the adaptation to change component with $\gamma_2 > \gamma_1$ for penalizing negative changes in trust score ($\alpha = 0.2;\ \beta = 0.8;\ \gamma_1 = 0.1;\ \gamma_2 = 0.4$). In this case, we see how the benefits of the history component – by slowly building trust from 0 to 1 – can be augmented by the adaptation to change component so that the drop in user behavior can be more quickly reflected in the overall trust rating.

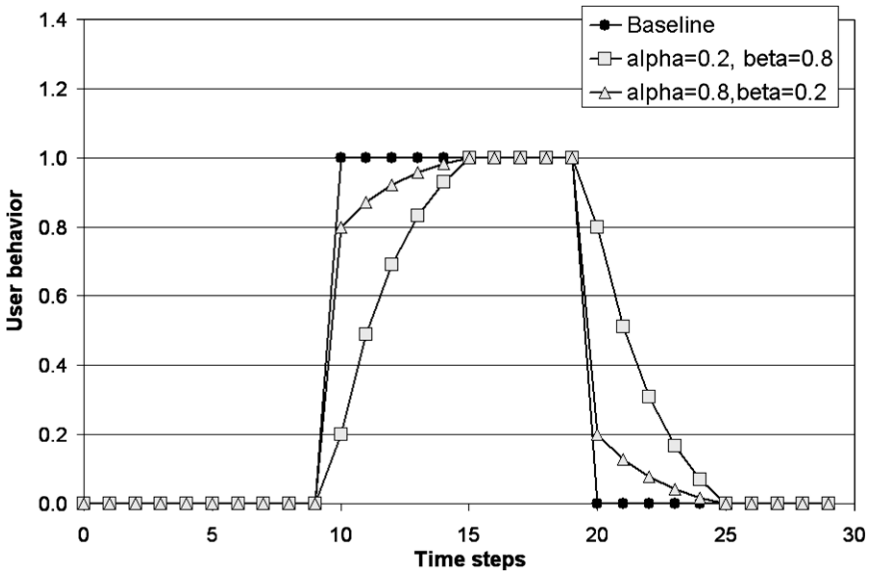

Fig. 1. Impact of history component on SocialTrust.

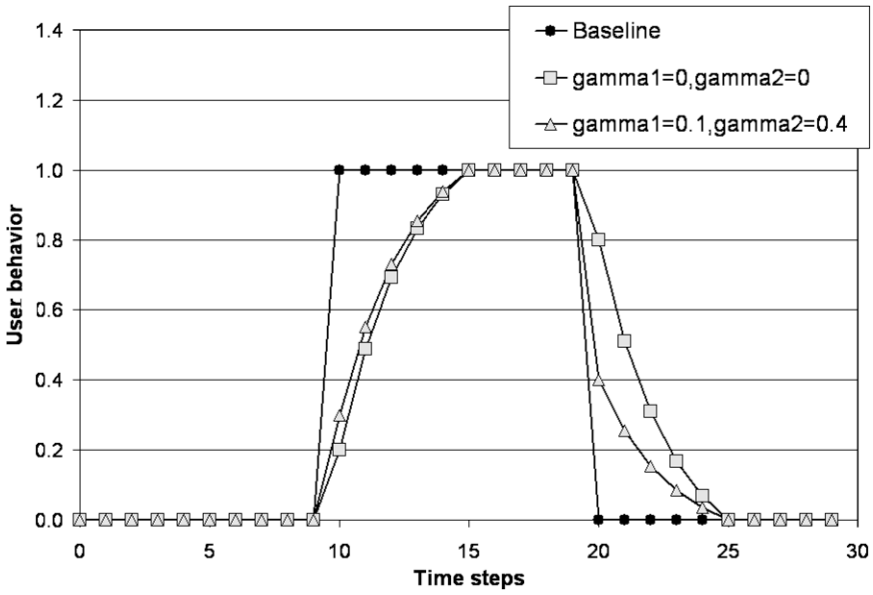

Fig. 2. Impact of change component on SocialTrust.

Of course the appropriate setting for these tunable knobs is application and scenario dependent. In this section, we have illustrated some of the possible factors that may impact these settings, but the optimal choice is an open question. In our ongoing research, we are deploying SocialTrust internally at Texas A&M to study these choices in more detail.

5. Personalized trust aggregation with mysocialtrust

In the trust framework so far, we have identified feedback ratings, user relationship quality, and the history and change components of trust as important features and seen how to assess trust based on each user's position in the social network. We have stressed the overall perspective of the community in assessing trust. In this section, we seek to balance the personal experiences of each user with the larger view of the entire community through a personalized extension to the trust framework called MYSOCIALTRUST that promotes a personalized version of the quality component of trust $Tr_q(i)$.

Suppose we are viewing the network from user *i*'s perspective. User *i* maintains his direct relationships and may have had some direct experience with a small fraction of all community members. In the current model, user *i* can assess the relative trustworthiness of an unknown user by relying on the community-wide aggregated trust values.

5.1. Basic personalization

One natural approach to provide a more personalized view over these global trust values is to combine the user's direct experiences with the global trust ratings via a linear combination:

$$Tr_q^{[i]}(j) = \alpha D^{[i]}(j) + (1-\alpha) Tr_q(j)$$

where we denote *i*'s direct trust rating of user *j* by $D^{[i]}(j)$ and the combined trust rating as $Tr_q^{[i]}(j)$. Regardless of how the personal trust ratings are made (e.g., personal trust ratings may be a function of direct experience or some trust propagation approach), the main drawback of this approach is its sparsity of data for augmenting the trust ratings of most users. Since any one user may have direct experience with only a small fraction of all community members, such a combined trust rating will result in the re-scoring of only a few other nodes.

5.2. Proximity-based personalization

One way to provide some personalization is to use the distance between nodes in the graph $\mathscr{SN}$ to weigh the global trust assignment. If we let $d(i,j)$ denote the shortest distance between two users in a social network (e.g., if user *i* links to user *k* who links to user *j*, then $d(i,j) = 2$). Discounting the global trust value of users who are far away in the network leads to the following personalized trust score:

$$Tr_q^{[i]}(j) = \alpha D^{[i]}(j) + (1-\alpha) Tr_q(j) \gamma^{d(i,j)}$$

where $\gamma (0 \leqslant \gamma \leqslant 1)$ is a discount factor for controlling how much to value distant users relative to close users. While such an approach to personalization has the advantage over the basic formulation by considering personalized node distance, it lacks the intuitive appeal of the random walk model in the original global quality component of trust.

5.3. Random walk personalization

Suppose instead that we augment the original SocialTrust random walk approach described in Eq. (4) to consider user *i*'s perspective. Replacing the global values $Tr_q(j)$, $R(j)$, and $F(j)$ with user *i*'s perspective (respectively $Tr_q^{[i]}(j)$, $R^{[i]}(j)$, and $F^{[i]}(j)$) we have:

$$Tr_q^{[i]}(j) = \lambda \sum_{k \in rel(j)} R^{[i]}(k) * Tr_q^{[i]}(k)/|rel(k)| + (1-\lambda) F^{[i]}(j) \tag{6}$$

We can interpret $Tr_q^{[i]}(j)$ as user *i*'s trust rating of user *j*. But how are the personalized user relationship quality and feedback ratings made in the first place? Since the trust rating is driven by these two inputs, it is of critical importance to identify how they are made.

One approach uses a similar spirit of the hop-based trust dampening suggested above to influence the user relationship quality and feedback ratings for each user. In this case we can replace $R^{[i]}(k)$ and $F^{[i]}(j)$ with estimates based on the global values and the distance of each user *j* from user *m*:

$$R^{[i]}(k) = R(k)\gamma^{d(i,k)}$$
$$F^{[i]}(k) = F(k)\gamma^{d(i,k)}$$

Plugging these values into Eq. (6) yields a recursive formulation in which the trust value of each user is impacted by the relative user relationship quality and feedback rating as a function of the user's distance from the point of view of user *i*. This formulation has the nice property of balancing the local and global perspectives relative to *i*. A user that is farther from *i* will require proportionately more high quality relationships (via the user relationship quality factor) to score as high as a user much closer to *i*. In this way, users that are deemed globally of high relationship quality and trust can score high even if they are quite distant from user *i*.

6. Evaluation

In this section, we evaluate the SocialTrust framework through simulations of community-based information sharing over real social network data. All decisions in the simulation study are based on the SocialTrust trust ratings – hence, we can see directly see if more robust information sharing is possible under a variety of parameter settings and user behaviors. We focus on three aspects: (i) a comparison of SocialTrust versus alternative trust models; (ii) the study of user relationship quality; and (iii) an evaluation of SocialTrust in the presence of strategies attempting to subvert its effectiveness, including clique

formation and collusive feedback. We find that the SocialTrust framework supports robust and tamper-resilient trust ratings even when large portions of the social network engage in behavior intended to undermine its effectiveness.

6.1. Experimental setup

Data: All of the experiments rely on data collected from MySpace, the largest social networking site and one of the few that provides open access to public user profiles. Many other sites (e.g., Facebook, LinkedIn) require a user account and, even then, access to the entire social network can be restricted.

We ran multiple parallel crawlers over MySpace in July 2006, beginning from a random sample of seed profiles. The crawlers followed the relationship links listed on each profile's front page in a breadth-first traversal of MySpace, resulting in a collection of 8,91,197 full-text profiles. Based on these profiles, we generated a directed graph consisting of 51,99,886 nodes representing both the collected full-text profiles and additional referenced profiles and 1,91,45,842 relationship links. A more detailed study of this dataset can be found in [12].

As a pre-processing step, we removed the default relationship links to the profile belonging to "Tom", the purported creator of MySpace who serves as a default friend for all new users, and whose super-user presence is atypical of most social networks. We performed validation of the graph, in which we confirmed that the link distribution of the graph follows a power-law and that the clustering coefficient (an indicator of the small-worldness of graphs) was 0.06. Both features have been observed to be important in social networks [23,38]. We also confirmed that the graph is connected.

Application scenario: As an application setting for evaluating the quality of SocialTrust, we consider a scenario in which an *originating user* has an information need (e.g., looking for a job in Texas, finding a good restaurant) for which she can use her social network. The basic scenario is this: a user browses her relationships up to some radius looking for candidate users to ask; based on an analysis of their profiles, she constructs a set of candidate users who might satisfy her information need; based on the provided trust ratings, she selects the top-k most trusted candidate users; she asks all top-k; if she is satisfied, she provides positive feedback to the trust manager; otherwise, she provides negative feedback.

Simulation setup: The simulation begins from a cold start, in which each user in the network is assigned a default trust score. Thereafter, users are randomly selected to begin a browsing session for a particular information need, they report their feedback to the trust manager, and at regular intervals the trust manager calculates the trust score for each user in the network for use in the next cycle. For each browsing session, we simulate a large browse over a relationship-based trust group with radius 7, in which the originating user browses using random selection with up to eight random neighbors selected at each step. We intentionally select such a large trust group (covering on average 26k users or 0.5 % of the network) to stress-test the quality of the trust values since more malicious users will be available to corrupt the quality of responses in each browsing session.

We model an originating user's information need using a simple unigram information retrieval model: a "query" term is randomly selected from the space of all MySpace profiles, weighted by the number of profiles in which it occurs. A profile encountered during browsing is considered a candidate based on a simple binary match between the selected term and the user's profile.

User behavior: We model two types of users: (i) malicious users, who always provide an irrelevant response when asked; and (ii) legitimate users, who sometimes accidentally provide an irrelevant response when asked.

Evaluation metric: For a query q, let R^+ denote the set of relevant users for q throughout the entire space of users and let R_n denote the n top-ranked candidate users (by trust value). We measure a focused version of the standard precision measure that considers the quality of the responses in the top-n (the relative precision @ n): $prec_n = \frac{|R^+ \cap R_n|}{min(|R_n|,n)}$. This relative precision metric measures the effectiveness of trust ratings by considering the quality of the top responses for a user's information need, even if fewer than n are returned. The traditional precision and recall measures provide little distinguishing power since malicious users may overwhelm an originating user with many poor quality responses. We measure the average performance over many browsing sessions starting from many different originating users, so we can identify system-wide quality metrics for comparing trust models.

Trust calculation: All trust calculations are performed using the Jacobi method for 25 iterations and a mixing parameter $\lambda = 0.85$. In all of our experiments, a simulation cycle consists of 5,000 browsing sessions. There are 30 simulation cycles in total. For each query, users provide feedback over the top-20 most trusted users they encounter. We report results over the last 5,000 browsing sessions, averaged over five simulation runs. In all of the reported experiments, we use the SocialTrust trust model described in Eq. (4). For the relationship quality component, we rely on the scoped random walk model with scope of $k = 3$ and an exponential correction factor with $\psi = 0.5$ and $\delta = 0.5$. We shall revisit some of these assumptions in the following experiments.

6.2. Comparing trust models

We first evaluate the quality of SocialTrust against alternative trust models and for varying degrees of user manipulation within the network. For each of the trust models, we consider seven scenarios: 10% of the network is malicious, 20%, ..., 70%. When asked, malicious users provide a corrupt (irrelevant) response with 100% probability; other users respond with a corrupt result with 5% probability. In all cases, if 100% of the network is malicious, the trust ratings are meaningless and the overall precision drops to 0.

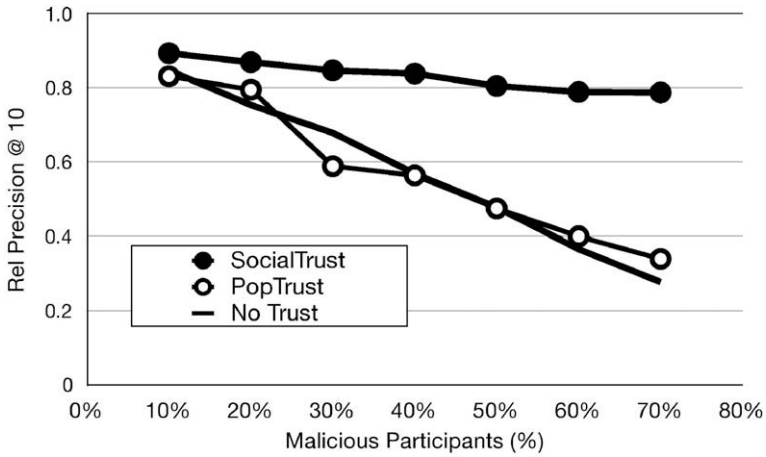

Fig. 3. SocialTrust vs. basic trust models.

In Fig. 3, we compare SocialTrust against the *No Trust* case – in which a user randomly selects among the users encountered in the browsing session – and the simple popularity-based *PopTrust* model presented in Section 2.3, Eq. (1). In both cases, we see that the relative precision for SocialTrust is resilient to the increase in malicious users, whereas the *No Trust* and *PopTrust* models degrade severely. With an increasing number of malicious users in the network, neither the *No Trust* model nor the *PopTrust* model gives the unsuspecting user any assurances as to the quality of the users in the network. At first glance, the fall in precision for the *PopTrust* model may be surprising, but consider that malicious users are distributed throughout the network at all ranges of popularity. Hence, when a proportion of the most popular (and most trusted) users behave maliciously, there is no mechanism for correcting this bad behavior.

Given that SocialTrust outperforms these naive models, how well does it perform against more sophisticated ones? In Fig. 4, we compare SocialTrust to the random walk trust model *RWTrust* presented in Section 2.3, Eq. (2). This type of random walk model has been popularized in the Web and P2P domains, and more recently adapted to social networks (where nodes are users and links are the relationships between users) as in [42] and [44].

We consider an *RWTrust* model that considers only the relationship structure of the social network; an *RWTrust + Feedback* model that uses feedback ratings as a priori trust (in a style similar to EigenTrust from the P2P domain and TrustRank in the Web domain); the preliminary SocialTrust [RQ Only] model that incorporates user relationship quality only but no feedback ratings (which is similar in spirit to credibility-based link analysis explored in the Web domain in [11]); and the final SocialTrust model.

First, both the *RWTrust* and *RWTrust + Feedback* models degrade severely, performing nearly as poorly as the naive *PopTrust* and *No Trust* approaches. At first glance, the fall in precision for these models may be surprising, but consider that malicious users are distributed throughout the network, meaning some of the initially most trusted users are malicious.

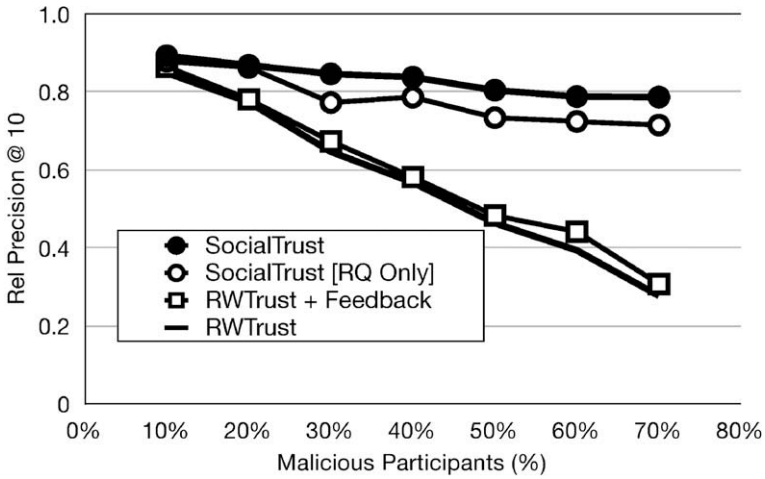

Fig. 4. SocialTrust vs. random walk trust models.

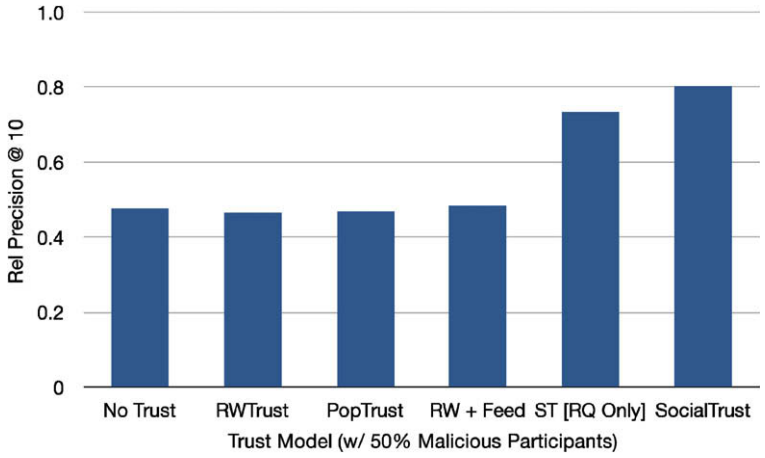

Fig. 5. Detail: comparing trust models.

When a proportion of these highly-trusted users behave maliciously, *RWTrust* and *RWTrust + Feedback* have no mechanism for correcting this bad behavior. In contrast, the SocialTrust model incorporates user relationship quality and feedback ratings into the trust assessment so that bad behavior is punished, and so the resulting precision measures are resilient to the presence of a large fraction of malicious users in the network. This is especially encouraging since the feedback ratings available in one simulation round may be incomplete for users who have not yet been rated in previous rounds. Also note that the inclusion of user relationship quality (SocialTrust [RQ Only]) provides the single biggest improvement in precision, since it reduces the influence of users who engage in poor quality relationships. When coupled together, both feedback ratings and user relationship quality provide the best performance (SocialTrust).

To further illustrate, we compare all of the trust models in Fig. 5 for the scenario when 50% of the network is malicious. Here, we can see the importance of considering user relationship quality (in the difference between SocialTrust [RQ Only] and the other random walk models), as well as the important but less significant impact of incorporating feedback ratings (in the difference between SocialTrust [RQ Only] and SocialTrust).

6.3. Impact of user relationship quality

Since the user relationship quality is such an important factor, we next compare several versions. We additionally consider the optimistic approach for $k = 1$ to $k = 5$, the pessimistic approach for $k = 1$ to $k = 5$ (with $\delta = 0.5$), as well as the exponential hop-based approach for $k = 1$ to $k = 5$. In Fig. 6, we report the relative precision @ 10 for the scenario when 50% of the users in the network are malicious, but with the different approaches for computing relationship quality incorporated into the trust model.

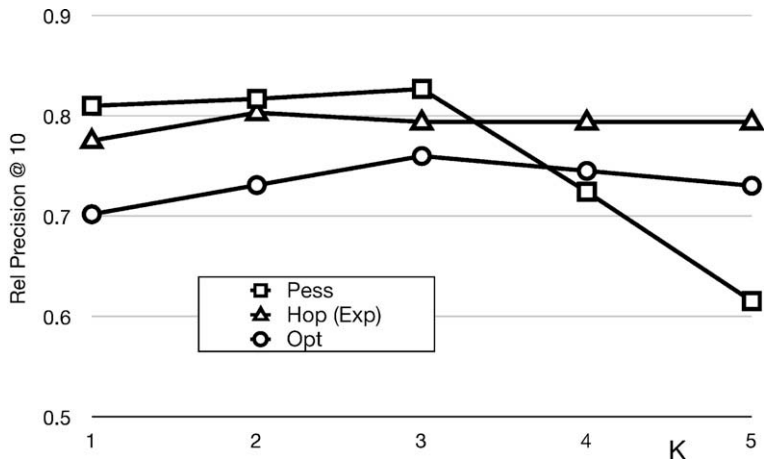

Fig. 6. Evaluating user relationship quality.

First, the optimistic and hop-based approach are stable and perform fairly well as the scope parameter k increases. These approaches penalize a candidate user's relationship quality score in proportion to the distance of malicious users from the candidate user. Direct relationships with malicious users result in a lower relationship quality score than paths of multiple hops to malicious users. In contrast, the pessimistic approach results in a worsening of precision as the scope increases. As k increases, most users have at least one path to a malicious user and are assigned a 0 or low relationship quality score. As the relationship quality score approaches 0 for nearly all users in the network, the rankings induced from the trust model become random, and so we see the precision fall considerably.

6.4. Clique formation

In our previous experiments, malicious nodes enter the network randomly. Suppose instead that malicious nodes seek to form cliques in the social network so that they can leverage their tightly-coupled relationship structure to overpower SocialTrust. Rather than randomly assigning users to be malicious, we now construct malicious cliques. The setup works like this: first a node is randomly selected and assigned to be a malicious node, then up to three-hops of its neighbors are also assigned to be malicious. We repeat this process until 10% of the network is malicious. This overall procedure continues for the 20% case, 30% case, up to the 70% case.

In Fig. 7 we report the relative precision @ 10 for SocialTrust over this clique-based strategy (*Clique*). As points of comparison, we also show the performance of SocialTrust over the original non-clique strategy (*Non-clique*), as well as the performance of the *No Trust* strategy over the clique-based strategy. Even in the presence of cliques, the SocialTrust approach provides resilient rankings as the fraction of malicious users increases. We attribute the success of the SocialTrust approach to its incorporation of user relationship quality, so that the influence of malicious cliques over the aggregated trust ratings is reduced.

6.5. Subverting feedback ratings

Suppose that in addition to providing irrelevant answers when asked, that malicious users also attempt to subvert the feedback ratings. So far, we have used the trust-aware restricted voting at the end of each simulation cycle, where a user's feedback is proportional to his trust rating. In this final experiment, we consider the other two voting schemes discussed in Section 4.1 – open voting and restricted voting.

Recall that the Trust-Aware approach allots a voting share to each user based on his trust value, so that more trusted users have greater sway over the feedback ratings of other users than do lowly trusted users. For the restricted voting case, each user is allotted an equal voting share for distributing among the users in his trust group who have answered its queries in the past. In the open voting case, there are no constraints on the number of votes cast by any user.

For each voting scheme, we assume that a malicious user always provides negative feedback for legitimate users, regardless of the quality of the answer provided; a legitimate user provides honest feedback. For the open voting case, we assume that malicious users ballot stuff the voting process, resulting in feedback ratings for legitimate users randomly distributed between [0,0.1]. Malicious users receive high feedback ratings randomly distributed between [0.9,1].

In Fig. 8, we compare the performance of the SocialTrust framework over each voting scheme. As the network tips over 50% malicious, the restricted voting case begins a steep decline. In this scenario, there are more malicious users in the

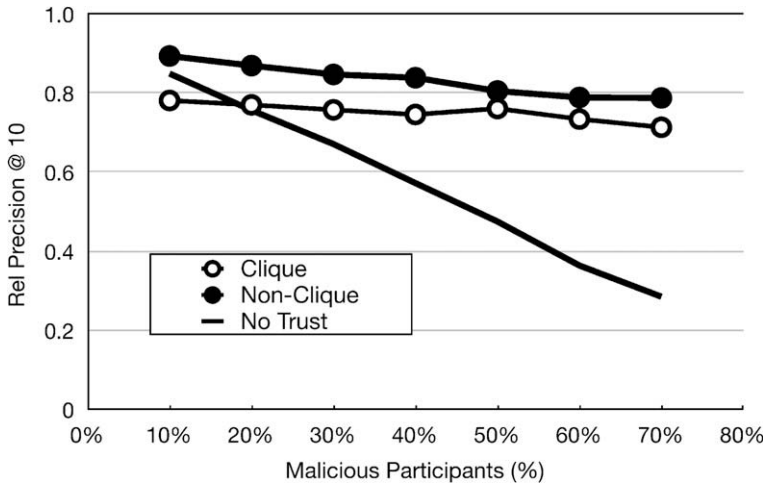

Fig. 7. Effectiveness of clique strategies.

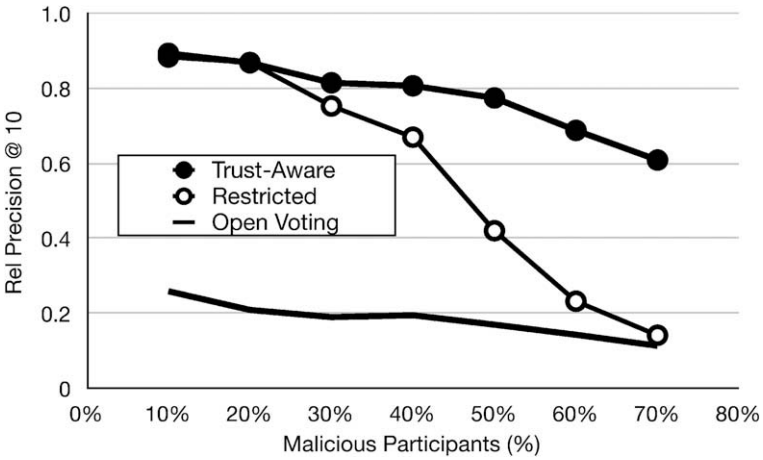

Fig. 8. Comparing feedback schemes.

network and so (regardless of their past trust values), they can promote other malicious users, so that in the following round these malicious users receive a boost in feedback rating (and hence, relationship quality, and ultimately, trust). For the open voting scheme, we see that precision is very low across the scenarios. Even a small percentage of malicious nodes can subvert the feedback ratings of legitimate users (and promote the scores of other malicious users), so that the derived trust ratings favor malicious users.

In contrast, the trust-aware voting scheme is fairly resilient; as more and more malicious users enter the network, the highly-trusted users manage to keep them under control. The robustness of the SocialTrust model, even with large portions of the network providing dishonest feedback, can be partially attributed to our model of how malicious users enter the network. In our simulations, malicious user are activated in 10% chunks. Since trust and feedback ratings are linked from round-to-round, the votes of legitimate users in one round can deter the malicious users from receiving high trust scores in the following round. In contrast, if 70% of the entire network were to suddenly behave maliciously, we would observe a steep degradation in precision. Based on this observation, we are studying additional feedback countermeasures to incorporate into future revisions of SocialTrust. This experiment emphasizes the need for high-quality feedback aggregation; without it, even the best trust model is subject to extreme manipulation.

7. Related work

The study of social networks has a rich history [28], and there has been great interest in modeling these networks and understanding how people efficiently use their social networks, e.g., [15,37,38]. The rise of online communities has spurred interest in community information management [14], social network formation [3], and the modeling and analysis of online social networks [2,23,25].

In this manuscript we consider the problem of *trust aggregation in online communities*, which can build on previous work on the important, but distinct, problem of assessing *direct trust*. Several studies have examined how to compute direct trust between nodes, including: [34], which developed statistical models of bid behavior on eBay to determine which sellers are suspicious; TrustGuard [36], which targeted strategically malicious P2P nodes who oscillate their behavior; PeerTrust[41], which studied P2P feedback mechanisms; [27], which stresses the need for direct trust; [16], which studies personalized trust and distrust propagation; and [43], which studied reputation formation in electronic communities. Note that direct trust can be interpreted differently depending on the context and the relevant community: for example, in eBay, trust is a measure of the fulfillment of a commitment; in a P2P network, trust is often a measure of file download success. Our goal is to propose a general framework for trust aggregation that can incorporate any of these direct approaches; in fact, elements of each of these direct trust approaches can be layered into the SocialTrust approach. Experimentally, we have grounded our evaluation in the specific context of community-based information sharing.

Research on trust and reputation in P2P networks (e.g., [1,6,13,7,26]) and on the Web (e.g., [18,40]) can inform the development of SocialTrust. Note that there are some key differences between these environments and social networks. For example, P2P networks often are concerned with high node churn and guaranteeing anonymity, and the networks are often formed via randomization protocols for establishing links between nodes. In contrast, online social networks tend to include long-lived profiles that strive to be known (i.e., are not anonymous), and links in the social network stress the personal connection. On the Web, users can rely on trust ratings over pages; typically, the user is divorced from the page-level trust assessment which often centers around hyperlink analysis. On social networks and in SocialTrust in particular, users are first-class participants in how trust is built and used.

8. Conclusion

We have presented the design and evaluation of the SocialTrust framework for aggregating trust in online social networks and provided the first large-scale trust evaluation over real social network data. The proposed framework supports tamper-resilient trust establishment in the presence of large-scale manipulation by malicious users, clique formation, and dishonest feedback. We have seen how trust group feedback, distinguishing between user relationship quality and trust, and tracking user behavior can result in more resilient trust ratings than in popular alternative random walk trust models similar to PageRank and TrustRank.

In our future work, we are interested in developing context-aware extensions of SocialTrust so that the network may support multiple trust views of each user depending on the context. We also see opportunities to augment the evaluation of user relationship quality, so that it considers more sophisticated features like the nature, duration, and value of each relationship. On the implementation side, we continue work on a SocialTrust-powered community platform that can be layered on top of existing social networks.

Acknowledgement

This work is partially supported by faculty startup funds from Texas A&M University and the Texas Engineering Experiment Station and by grants from NSF CyberTrust, NSF CSR, NSF ITR, an IBM faculty award, IBM SUR grant, and an AFOSR grant.

References

[1] K. Aberer, Z. Despotovic. Managing trust in a Peer-2-Peer information system, in: CIKM, 2001.
[2] L.A. Adamic, E. Adar, How to search a social network, Social Networks 27 (3) (2005) 187–203.
[3] L. Backstrom, D.P. Huttenlocher, J.M. Kleinberg, X. Lan, Group formation in large social networks, in: KDD, 2006.
[4] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, Z. Su, Optimizing web search using social annotations, in: WWW, 2007.
[5] S.B. Barnes, A privacy paradox: social networking in the United States, First Monday 11 (9) (2006). September.
[6] E. Bertino, E. Ferrari, A.C. Squicciarini, Trust-x: a peer-to-peer framework for trust establishment, IEEE Transactions on Knowledge and Data Engineering 16 (7) (2004) 827–842.
[7] C. Boyd, Teenagers used to push Zango on MySpace, http://www.vitalsecurity.org, 2006.
[8] D. Boyd, Social network sites: Public, private, or what? The Knowledge Tree, 2007.
[9] C.R. Brooks, N. Montanez, Improved annotation of the blogosphere via autotagging and hierarchical clustering, in: WWW, 2006.
[10] C. Cahoon, Facebook phonies, The Buchtelite, University of Akron, December 2005.
[11] J. Caverlee, L. Liu, Countering web spam with credibility-based link analysi, in: PODC, 2007.
[12] J. Caverlee S. Webb, A large-scale study of MySpace: observations and implications for online social networks, in: Second International Conference on Weblogs and Social Media (AAAI), 2008.
[13] F. Cornelli, E. Damiani, S.D. Capitani, Choosing reputable servents in a P2P network, in: WWW, 2002.
[14] A. Doan, R. Ramakrishnan, F. Chen, P. DeRose, Y. Lee, R. McCann, M. Sayyadian, W. Shen, Community information management, IEEE Data Engineering Bulletin (March) (2006).
[15] P.S. Dodds, R. Muhamad, D.J. Watts, An experimental study of search in global social networks, Science 301 (5634) (2003) 827–829.
[16] R. Guha, R. Kumar, P. Raghavan, A. Tomkins, Propagation of trust and distrust, in: WWW, 2004.
[17] Z. Gyöngyi H. Garcia-Molina, Link spam alliances, in: VLDB, 2005.
[18] Z. Gyöngyi, H. Garcia-Molina, J. Pedersen, Combating web spam with TrustRank, in: VLDB, 2004.
[19] T.N. Jagatic, N.A. Johnson, M. Jakobsson, F. Menczer, Social phishing, CACM 50 (10) (2007) 94–100.
[20] S. Kamvar, B. Yang, H. Garcia-Molina, Secure score management for peer-to-peer systems, Technical Report, Stanford University, 2004.
[21] S.D. Kamvar, M.T. Schlosser, H. Garcia-Molina, The EigenTrust algorithm for reputation management in P2P networks, in: WWW, 2003.
[22] G. Koutrika, F.A. Effendi, Z. Gyöngyi, P. Heymann, H. Garcia-Molina, Combating spam in tagging systems, in: AIRWeb'07: Proceedings of the Third International Workshop on Adversarial Information Retrieval on the Web, 2007.
[23] R. Kumar, J. Novak, A. Tomkins, Structure and evolution of online social networks, in: KDD, 2006.
[24] R. Li, S. Bao, Y. Yu, B. Fei, Z. Su, Towards effective browsing of large scale social annotations, in: WWW, 2007.
[25] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, A. Tomkins, Geographic routing in social networks, Proceedings of the National Academy of Sciences 102 (33) (2005) 11623–11628.
[26] S. Marti, H. Garcia-Molina, Taxonomy of trust, Computer Networks 50 (4) (2006) 472–484.
[27] P. Massa, P. Avesani, Controversial users demand local trust metrics, in: AAAI, 2005.
[28] S. Milgram, The small-world problem, Psychology Today (March) (1967) 60–67.
[29] R. Monastersky, The number that's devouring science, The Chronicle of Higher Education, October 2005.
[30] E. Nussbaum, Kids, the internet, and the end of privacy, New York Magazine, February 2007.
[31] H. Ozbay, Introduction to Feedback Control Theory, CRC Press Inc., 1999.
[32] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank citation ranking: bringing order to the Web, Technical Report, Stanford University, 1998.
[33] M. Richardson, R. Agrawal, P. Domingos, Trust management for the semantic web, in: ISWC, 2003.
[34] S. Rubin, M. Christodorescu, V. Ganapathy, J.T. Giffin, L. Kruger, H. Wang, An auctioning reputation system based on anomaly detection, in: CCS, 2005.
[35] M. Sanchez. Pranksters posting fake profiles on MySpace. http://www.dfw.com/, 2006.
[36] M. Srivatsa, L. Xiong, L. Liu, TrustGuard: countering vulnerabilities in reputation management for decentralized overlay networks, in: WWW, 2005.
[37] S. Wasserman, K. Faust, Social Network Analysis, Cambridge University Press, Cambridge, 1994.
[38] D.J. Watts, Networks, dynamics, and the small world phenomenon, American Journal of Sociology 105 (2) (1999) 493–527.
[39] D.J. Watts, P.S. Dodds, M.E.J. Newman, Identity and search in social networks, Science 296 (2002) 1302–1305.
[40] B. Wu, V. Goel, B. Davison, Topical TrustRank: using topicality to combat web spam, in: WWW, 2006.
[41] L. Xiong, L. Liu, Supporting reputation-based trust for P2P electronic communities, TKDE 16 (7) (2004).
[42] S.A. Yahia, M. Benedikt, P. Bohannon, Challenges in searching online communities, IEEE Data Engineering Bulletin (2007).
[43] B. Yu, M.P. Singh, A social mechanism of reputation management in electronic communities, Cooperative Information Agents (2000).
[44] J. Zhang, M.S. Ackerman, L. Adamic, Expertise networks in online communities: structure and algorithms, in: WWW, 2007.