A Trust-Aware System for Personalized User Recommendations in Social Networks

Magdalini Eirinaki, Malamati Louta, Iraklis Varlamis

***Abstract*—Social network analysis has recently gained a lot of interest because of the advent and the increasing popularity of social media, such as blogs, social networking applications, micro-blogging, or customer review sites. In this environment, trust is becoming an essential quality among user interactions and the recommendation for useful content and trustful users is crucial for all the members of the network. In this work, we introduce a framework for handling trust in social networks, which is based on a reputation mechanism that captures the implicit and explicit connections between the network members, analyzes the semantics and dynamics of these connections and provides personalized user recommendations to the network members.**

***Index Terms*—social networks, recommendation, personalization, reputation, trust**

I. Introduction

Social network analysis has been a major area of research for sociologists for many years. Recently, it has gained a lot of interest with the advent of Web 2.0 and the enormous increase in the use of social networking applications, customer review sites, blogs, wikis etc. Such media present features unique to the Web, in terms of shared authorship, multitude of user-provided tags, inherent connectivity between users and their posted items and high update rate. All these characteristics could be exploited in order to mine interesting information about the dynamics of users' interactions.

One common type of analysis is the identification of communities of users with similar interests [1], [2]. Another research direction is the identification of content that could be of potential interest, whether this is a product review, a blog or a tweet. Collaborative filtering is the most broadly adopted technique used to predict future item ratings based on the user's past behavior as well as ratings of other similar users. It has been shown that incorporating social network relationships (e.g., friendship) and respective opinions/ratings improves the prediction, and consequently the recommendation process [3], [4], [5]. A similar line of work focuses on content ranking, which is consequently employed to recommend the top-ranked items (reviews, blogs, comments, tweets, etc) to users. This is particularly important since the rapid increase in terms of content and users of social media shifts the problem of information search to that of information discovery. The largest body of work in this area generates overall rankings [6], [7], [8] and only recently there have been some efforts in personalizing the ranking [9] and in providing different rankings depending on the scope under which the network is examined [10], [11], [12].

Recently, trust has been introduced in the context of recommender systems for social networks [13], [14], [15]. Trust in general is a multi-faceted concept: it is subjective and non-symmetric [16], dynamic and context-specific [17], while it is often defined as the belief of an entity in the benevolence of another entity to act honestly and reliably in opposition to distrust [18].

This work proposes a trust-aware system for personalized user recommendations in social networks. Contrary to the initial works on user recommender systems for social networks that do not incorporate trust [19], [20], [21], and following the paradigm of more recent research works [22], [23], [24], [25], our work capitalizes on trust (and distrust) between people in order to assist members of a community to make decisions about other members of the same community (e.g., an online social network, the blogosphere, a social bookmarking application etc.). More specifically, the proposed system provides users with personalized positive and/or negative recommendations which can be used to establish new trust/distrust connections in the social network. Hereafter, we assume that the notion of trust captures both the user's social context (e.g., friends, enemies) expressed through explicit user-to-user connections, as well as users' common interests and desires inferred from explicit and implicit user-to-item connections.

The proposed recommender system is based on a *reputation mechanism* that rates participants using observations, past experiences, and other user's view/opinion. In order to compute the reputation of each member, we adopt several properties of trust such as, *transitivity, personalization*, and *context* [26] and draw ideas from sociology axioms [27]. Trust is not perfectly transitive in social networks, in that trust decays along the transition path, but it is generally agreed that it can be communicated between people [22], [23], [28], [29]. Trust is also personalized in that it is subjective and affected by each user's personal beliefs, as well as those of members whom the user respects and trusts. Additionally, in order to address the social network dynam-

M. Eirinaki is with the Computer Engineering Department, San Jose State University, San Jose, CA, USA.

M. Louta is with the Department of Informatics and Telecommunications Engineering, University of Western Macedonia, Kozani, Greece

I. Varlamis is with the Department of Informatics and Telematics, Harokopio University, Athens, Greece

ics, we have incorporated in our system the element of time. To this direction, we suggest that reputation fades by time, thus the positive (negative) reputation value of a user tends to zero unless new explicit or implicit trust (distrust) and liking (disliking) statements are added frequently. Finally, we assume that the context of trust is the same among community members.

In a nutshell, our contribution is a system for providing personalized user recommendations. We exploit positive and negative, time-dependent trust-related information, expressed either explicitly or implicitly. We propose a collaborative reputation mechanism which captures and quantifies the users' connections and capitalizes on trust propagation and on the dynamics of the social network. Using this mechanism, the system proposes new trust/distrust connections to the network's members. We should point out that the system can be applied to any type of social network, even in the absence of explicit trust connections, since in such cases only the implicit expressions of trust will be considered for the ranking and recommendation of users.

The paper is organized as follows. In Section II we provide an overview of the related research literature and discuss the contribution of this study. We present the fundamental concepts of the trust-aware recommendation system in Section III and provide the mathematical formulation of the user reputation rating system in Section IV. We evaluate the proposed system in Section V and conclude with an outline of our future plans in Section VI.

II. Related Work

The analysis of content and links in social networks has gained a lot of momentum, resulting in an increase of research in the related fields. In what follows we examine related work in the areas covered by our system, namely trust and trust propagation, time dynamics and negative trust, with an emphasis on the works that generate user recommendations. Even though the reputation mechanism is an integral part of our system, due to space limitations we omit a discussion on the related work since our main focus is on the system's characteristics mentioned above.

The largest body of work involving positive trust and/or trust propagation in the context of recommender systems has focused on item recommendations [13], [14], [28], [29], [30], [31], [32], [33]. Time dynamics have been introduced by Walter *et al.* [34], [15]. The notion of trust propagation through transitivity is employed, and, similarly to our work, "discounting takes place by multiplying trust values along paths". This work has several common aspects to our approach in terms of modeling the trust propagation and dynamics, however, the model assumes only positive trust and aims at generating item recommendations.

The problem of user recommendations in social networks, initially formulated as a link prediction problem [35], has recently gained a lot of momentum. In their works, Chen *et al.* [19] and Guy *et al.* [20], [21] propose several algorithms, based on different combinations of content similarity, social link information, and common items (e.g., common publications) among users in order to recommend new friends to the users of a social network. The element of trust among users is not incorporated, and the proposed model is only applicable to social networking applications and not other social media.

In the case of blogs, several ranking algorithms have been suggested that exploit explicit [6] and/or implicit [7], [8] hyperlinks between blogs. These hyperlinks can be regarded as indications of positive trust among bloggers and the models generate a ranking that can be used for blog recommendations. A similar effort that also incorporates the content when ranking tweets is presented by Weng *et al.* [12]. All the aforementioned approaches can be regarded as cases of user recommender systems (since blogs or tweets are usually originated by a single user), but are very specific to the characteristics of each medium.

A more generic model, that can be readily applied to any social medium, has been presented in our previous work [24], [25]. We defined both local and global metrics for user recommendations in social media that could be applied to any social media. However, in that work, we did not incorporate the notion of negative trust among users.

Negative trust, previously introduced in different contexts such as peer-to-peer networks, web recommender systems, and community discovery [2], [30], [36], [37], [38], has recently been introduced in the context of user recommendations in social networks [22], [23]. Kunegis *et al.* [22] focus on predicting unpopular users and the sign of links using the Slashdot network as their testbed. They employ signed variants of global network characteristics such as the clustering coefficient, node-level characteristics such as centrality and popularity measures, and link-level characteristics such as distances and similarity measures. The experiments demonstrated the multiplicative transitivity of trust and supported the idea that "the enemy of my enemy is my friend". On the contrary, Leskovec *et al.* [23] who try to predict positive and negative links in social networks using a machine-learning framework and ideas drawn from sociology have derived opposite results. Both works are very similar to ours in that they incorporate the notion of negative trust relationships in order to generate user recommendations in a social network. However, the work of Kunegis *et al.* is dependent on the idiosyncrasies of the specific network they are analyzing. The work of Leskovec *et al.* , while being generic, has a slightly different focus – that of predicting positive or negative edges (i.e. relations) between users. Moreover, none of the above works considers time and its effect on trust.

Our work touches all the aforementioned areas of research, yet is novel in several ways. Our focus is on personalized *user recommendations* exploiting both *positive and negative trust relationships*. The trust of a user to another user is based on a personalized reputation rating, which quantifies explicit connections among users (e.g., friendship, trust, or distrust) and implicit connections inferred from the interactions among users (e.g., comments, like and dislike statements, etc.). Additionally, our model supports *trust propagation* through explicit user connections in the

social network. Moreover, combining the merits of our previous work on social network dynamics [9], [10], we incorporate the element of *time* in the calculation of the users' reputation. Finally, the proposed model is *generic*, in that it can he readily applied to any type of social network, including blogs, social networking applications, microblogging sites, etc.

III. Recommender System

This work proposes a trust-aware system for providing personalized user recommendations to the members of a social network in an efficient manner based on a robust reputation management model. Specifically, after processing information published on the network, connections (both explicit and implicit) that bear trust semantics between members are formed (phase 1), reputation ratings are estimated (phase 2) and personalized recommendations (both positive and negative) are generated (phase 3). These recommendations are the basis for creating new trust and/or distrust connections in the social network.

In what follows we elaborate on the fundamental aspects of the three phases identified above. Specifically, we describe in detail the trust connections that may be identified in social networks, the reputation rating formation process and the recommendation generation engine.

Phase 1: User Connection Formation. Our system differentiates between explicit trust/distrust bonds amongst users that carry strong trust semantics and implicit trust statements that form more transient user connections in the network. A user may explicitly state his/her trust/distrust on another user or may express it implicitly through his/her opinion (e.g., a "Like", a comment) on another user's published content item. Trust connections may be categorized in four distinct categories, namely, a) Explicit User-to-User connections, b) Explicit User-to-Item connections, c) Implicit User-to-Item Connections and d) Implicit User-to-User Connections.

Explicit User-to-User connection. A user may explicitly relate to another user by forming trust or distrust connections. Such connections represent more permanent bonds between users (e.g., a friendship or collaboration in the real world). For example, users can trust/distrust other users in Epinions, while they can tag users as friends/foes in Shashdot zoo. We model this profile data using trust or distrust links between users. We also assume that each network member maintains and updates two lists: a "friend" and an "enemy" list containing his/her trusted and untrusted users respectively. The list of friends comprises members that the user already trusts or can trust and interact with in the future. The main idea behind the list of enemies is that it comprises members who have received many negative trust scores by the user, his fellows or other members of the network (depending on the model) and are deemed untrustworthy for the user. The social networking service can use this list in order to alert the user when an enemy attempts to interact with him/her.

Explicit User-to-Item connection. In this type of connection, the user provides a "Like" or "Dislike" type of comment to a specific item published by another user. The semantics of opinion expression differ among applications. The comment can be, for instance, a thumbs-up or thumbs-down tag (as in the case of posts in social networking applications), or a positive or negative rating (as in the case of customer reviews in a product reviewing site), and carries no textual content and usually no timestamp information.

Implicit User-to-item connection. A slightly different type of connection inside a social network is the implicit User-to-Item connection, which is implemented through content items. Each content item published by a user has a unique identifier and a timestamp, and may contain one or more hyperlinks that point to other content items inside the social network or items (URLs) on the web. Preference to an item is shown implicitly, for example by sharing an article in Reddit or Facebook, by retweeting a post in Tweeter, by positively or negatively commenting on a user's post, etc.

Implicit User-to-User connection. Explicit and implicit User-to-Item connections from a user to the items of another user can be employed in order to infer the implicit User-to-User connection between the two users. The User-to-Item information is mapped to the User-to-User level and is aggregated in order to provide a single implicit User-to-User connection.

At this point it should be noted that although distrust connections are not supported in all social networks, we include them in our model, since they are very important for the management of trust.

Phase 2: Reputation Rating Estimation. The reputation mechanism quantifies the trust connections identified in the social network and provides personalized ratings expressing the local belief of a user (hereafter, referred to as the "evaluator" user) with respect to other members of the network (hereafter, referred to as "target" users). Reputation ratings are collectively formed, incorporating the evaluator's own view on the target user as well as the opinion of a number of other members of the social network (hereafter, referred to as "witnesses") with respect to the user under evaluation. The users' referral network (i.e., set of witnesses) is formed within specific circles of trust and distrust from the evaluator user based on concepts drawn from sociology. Specifically, we consider in a breadth-first search fashion the opinion of the evaluator's friends (i.e., users in the evaluator's "friend list"), and the opinion of the evaluator's "enemies" (i.e., users in the evaluator's "enemy list").

For example, in Figure 1 user u_j is the evaluator, user u_i is considered as a target user and u_q is a witness who shares with u_j his/her beliefs for u_i. Trust and distrust can be expressed with discrete positive and negative reputation values (e.g., +1 and -1), or by real values in the same range. A zero value denotes the absence of a connection between two users.

In order for an evaluator user to form and consequently

update his/her reputation rating with respect to a target user, our model takes into account the explicit and implicit connections of the evaluator to this target member formed during a specific time period. It then aggregates the more recent user ratings (i.e., the user ratings estimated during the more recent time periods), and provides the *local rating* assigned to the target member by the evaluator. Taking into account both the local rating of the evaluator (expressing the evaluator's own view on the target user) as well as the local ratings of a number of witnesses (expressing their own trust-related experiences), the model forms the *collaborative rating*.

The proposed reputation rating mechanism captures the effect of time (e.g., freshness of links) by modeling the fact that more recent events (i.e., newly added explicit or implicit trust(distrust) and like (dislike) statements) should weigh more in the evaluation of the target user's overall reputation rating by the evaluator. The use of time information allows to distinguish between users who attain a high reputation for a short time period and users who manage to maintain their reputation at a constantly high level. Thus, the social network's dynamic aspect is taken into account and is effectively addressed.

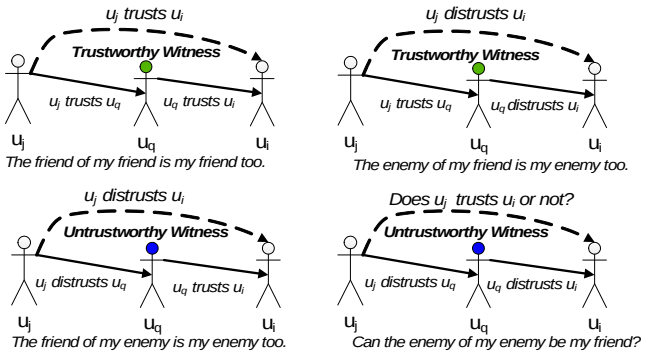

Fig. 1. Transitivity of positive and negative trust statements

Phase 3: Recommendations Generation. Based on the overall reputation ratings of the social network members as assessed by the evaluator user, the proposed system generates personalized positive and/or negative user recommendations, which can be used to form new trust and/or distrust connections. Positive recommendations can be used from the members in order to connect to new people (in social networking sites), subscribe to new blogs (in the blogosphere), share resources (in social bookmarking applications), etc. On the other hand, in the case of negative recommendations, the model in essence generates a list of untrustworthy users. This personalized "blacklist" can be exploited by the recommender system in order to alert users when content items are published from such untrustworthy users and discourage them from linking or browsing such content, or filter it out from their content feed. Both types of recommendations could be exploited in order for a user to update his/her trust and distrust connections in the social network.

IV. User Reputation Rating System Formulation

Let us assume the presence of N users $U = \{u_1, u_2, ...u_N\}$ in a social network. Every member $u_j \in U$, publishes several content items whilst in the network. Additionally, $F(u_j)$ and $E(u_j)$ denote the "friend list" and the "enemy list" maintained by user u_j, respectively.

A. Local Rating

The suggested model assumes that the local rating estimation takes place at consecutive, equally distributed time intervals denoted henceforth as t_k, $k \in \mathbb{N}$.

The user reputation rating $Rating(u_j \rightarrow u_i, t_k)$ of u_i from u_j at time period t_k is given by the following formula:

$$\begin{aligned} Rating(u_j \rightarrow u_i, t_k) = w_{user} \cdot UserConn(u_j \rightarrow u_i, t_k) \\ + w_{expl} \cdot ExplConn(u_j \rightarrow u_i, t_k) \\ + w_{impl} \cdot ImplConn(u_j \rightarrow u_i, t_k) \end{aligned} \quad (1)$$

where $w_{user} + w_{expl} + w_{impl} = 1$.

As may be observed from Equation 1, the rating of target u_i is a weighted combination of three factors. The first factor corresponds to the explicit User-to-User trust/distrust connections. It has been assumed that $UserConn(u_j \rightarrow u_i, t_k)$ lies within the [-1,1] range, where a value close to 1(-1) indicates that the target u_i is a friend (enemy) of the evaluator user u_j. The factor $UserConn(u_j \rightarrow u_i, t_k)$ can be modeled as a binary decision variable taking values 1 or -1 or take any value in the [-1,1] range providing a rating of the "friends" or "enemies" in the two lists.

The second factor, $ExplConn(u_j \rightarrow u_i, t_k)$, corresponds to the explicit User-to-Item connections as expressed by comments of user u_j to content items published by u_i at time period t_k. This factor has been assumed to lie within the [-1,1] range and is defined as follows:

$$\begin{aligned} ExplConn(u_j \rightarrow u_i, t_k) = \\ \frac{PosExpl(u_j \rightarrow u_i, t_k) - NegExpl(u_j \rightarrow u_i, t_k)}{PosExpl(u_j, t_k) + NegExpl(u_j, t_k)} \end{aligned} \quad (2)$$

where $PosExpl(u_j \rightarrow u_i, t_k)$ and $NegExpl(u_j \rightarrow u_i, t_k)$ denote the number of positive and negative User-to-Item explicit opinions, respectively (i.e., "Like" and "Dislike") as expressed by user u_j, at time period t_k, on the content items published by user u_i, and $PosExpl(u_j, t_k) + NegExpl(u_j, t_k)$ denotes the total number of opinions expressed by user u_j in time period t_k on any published content item. At this point, it should be noted that if no timestamp information is available, then Equation 2 takes into account all the expressed opinions, without any time-related restrictions.

In a similar manner, the third factor $ImplConn(u_j \rightarrow u_i, t_k)$ corresponds to the implicit User-to-Item connections and depends on the number of links from the content items published by user u_j at time period t_k on the content items published by user u_i. A link from a content item published by user u_j at time period t_k on a content item published by user u_i denotes the temporary interest (i.e., during time

period t_k) of user u_j to the ideas of user u_i. This interest may be positive, meaning that user u_j supports the idea expressed, or negative, meaning that user u_j disagrees with the published content item. This factor also lies within the [-1,1] range and is given by the following equation:

$$ImplConn(u_j \rightarrow u_i, t_k) = \frac{PosImpl(u_j \rightarrow u_i, t_k) - NegImpl(u_k \rightarrow u_i, t_k)}{PosImpl(u_j, t_k) + NegImpl(u_j, t_k)} \quad (3)$$

where $PosImpl(u_j \rightarrow u_i, t_k)$ and $NegImpl(u_j \rightarrow u_i, t_k)$ denote the number of positive and negative User-to-Item implicit connections, as expressed by links from the content items published by user u_j at time period t_k on the content items published by user u_i, respectively, and $PosImpl(u_j, t_k) + NegImpl(u_j, t_k)$ denotes the total number of links (expressing both positive and negative interest) from the content items published by user u_j in time period t_k on any published content item.

Weights w_{user}, w_{expl} and w_{impl} provide the relative significance of the three factors (i.e., User-to-User connections, User-to-Item explicit connections and User-to-Item implicit connections respectively). From the aforementioned analysis, it is obvious that $Rating(u_j \rightarrow u_i, t_k)$ in Equation 1 lies within the [-1,1] range.

For the formation of the local user reputation rating at the *current* time period t_c, the evaluator considers only the r more recent ratings formed by the user. The value of r determines the memory of the system. Small values of r mean that the memory of the system is short, whereas large values consider a longer memory for the system. The local reputation rating $LocalRating(u_j \rightarrow u_i, t_c)$ of user u_i, as estimated by u_j at time period t_c is defined as follows:

$$LocalRating(u_j \rightarrow u_i, t_c) = \sum_{\substack{k=c-r+1 \\ k>0}}^{c} df_k \cdot Rating(u_j \rightarrow u_i, t_k) \quad (4)$$

where $Rating(u_j \rightarrow u_i, t_k)$ denotes the user rating attributed to target user u_i by the evaluator user u_j at time period t_k as described above and the discount factor df_k provides the relative significance of the $Rating(u_j \rightarrow u_i, t_k)$ factor estimated at time period t_k to the overall u_i rating estimation by the evaluator u_j. The weight df_k is normalized ($\sum_{k=c-r+1,k>0}^{c} df_k = 1$) and defined as follows:

$$df_k = \frac{f_k}{\sum_{l=1}^{r} f_l} \quad (5)$$

where $f_k = \begin{Bmatrix} t_{r-c+k}, & c \geq r \\ t_k, & c < r \end{Bmatrix}$.

In essence, the discounting factor df_k decays with time, allowing for more recent ratings to receive much higher weight than older ones.

B. Collaborative Rating

As previously discussed, users in a social network form their opinion on other users based on their personal beliefs or interests as well as the opinions of other users, who act as witnesses. In order to estimate the rating of a target user u_i, the evaluator user u_j needs to contact a set $W(u_j \rightarrow u_i)$ of Q witness users ($U_q \in W(u_j \rightarrow u_i)$, $q \in [1..Q]$) in order to get feedback reports on the performance of u_i. The overall collaborative rating $CollRating(u_j \rightarrow u_i, t_c)$ of target user u_i is estimated by the evaluator user u_j at the current time period t_c using the following formula:

$$CollRating(u_j \rightarrow u_i, t_c) = cred(u_j \rightarrow u_j, t_c) \cdot LocalRating(u_j \rightarrow u_i, t_c) + \sum_{\substack{q=1 \\ q \neq i,j}}^{Q} cred(u_j \rightarrow u_q, t_c) \cdot LocalRating(u_q \rightarrow u_i, t_c) \quad (6)$$

As may be observed from Equation 6, the collaborative rating of the target user u_i is a weighted combination of two summands; the first is based on the direct experiences of the evaluator user u_j, while the second represents the rating of u_i as contributed by the Q witnesses.

The weight $cred(u_j \rightarrow u_q, t_c)$ is a measure of the credibility of witness u_q and the respective rating of u_i in the eyes of the evaluator u_j. In the context of this study it is expressed as a function of the local rating attributed to each witness u_q by the evaluator u_j. Specifically, considering only as witnesses the users who are explicitly connected to the evaluator user (i.e., friends and enemies), $cred(u_j \rightarrow u_q, t_c)$ is given by the following equation:

$$cred(u_j \rightarrow u_q, t_c) = \frac{LocalRating(u_j \rightarrow u_q, t_c)}{\sum_{u_q \in W(u_j \rightarrow u_i) \cup u_j} |LocalRating(u_j \rightarrow u_q, t_c)|} \quad (7)$$

where $LocalRating(u_j \rightarrow u_q, t_c)$ is the local rating attributed to witness u_q by the evaluator u_j(note that $LocalRating(u_j \rightarrow u_j, t_c) = 1$). It may be easily concluded that weights $cred(u_j \rightarrow u_q, t_c)$ fall in the range [-1,1].

C. Transitivity of trust

As already described, in order to estimate the collaborative user reputation rating, the evaluator contacts a set of witnesses in order to get feedback reports on the users' performance. Witnesses may be categorized in four distinct categories, namely a) friends of friends, b) enemies of friends, c) friends of enemies and d) enemies of enemies, as depicted in Figure 1.

Friends of friends. The first category comprises users who are members of user's u_j "friend list" $F(u_j)$ (depth=1), or are friends of the friends of u_j, thus, being members of users' u_q ($u_q \in F(u_j)$) "friend list" $F(u_q)$ (depth=2). According to the sociology axiom "the friend of my friend is my friend" [27] and experimental results

in online social networks [23], positive trust can be safely propagated in a wider transitivity horizon (depth > 2).

Enemies of friends. The second category comprises users who are enemies of the friends of the evaluator user u_j (thus, they are members of users' u_q ($u_q \in F(u_j)$) "enemy list" $E(u_q)$ (depth=2). For depth > 2 we can safely talk only for the friends of "enemy" users in the previous list. The intuition lies behind the axiom "the enemy of my friend is my enemy" and consequently all friends of "my enemy" (i.e., in deeper levels) are also enemies.

Friends of enemies. The third category comprises the direct enemies of the evaluator user u_j (thus, they are members of user's "enemy list" $E(u_j)$ (depth=1)), as well as those being friends with the enemies of the user (thus, they are members of users' u_q ($u_q \in E(u_j)$) "friend list" $F(u_q)$ (depth=2). For depth > 2, we again can safely talk only for the users in the "friend list" of users of previous lists, who are considered enemies of u_j. The intuition lies behind the axiom "the friend of my enemy is my enemy" and consequently all friends of "my enemy" (i.e., in deeper levels) are also enemies.

Enemies of enemies. Finally, the fourth category comprises users being enemies of the enemies of the evaluator user u_j (thus, they are members of users' u_q ($u_q \in E(u_j)$) "enemy list" $E(u_q)$ (depth=2)). As it is experimentally shown in [23], we cannot draw safe conclusions on whether these users are friends or enemies of the evaluator user u_j.

The first category is expected to contribute significantly to the generation of positive recommendations (the opinion of the friend of one's friend etc. in general coincides with his/her own view), while quite the opposite stands for the second and third categories, which are expected to contribute significantly to the generation of negative recommendations (the opinion of the friends of one's enemies in general is different from his/her own view). Finally, the last category seems to raise a controversial issue, as there are contradicting opinions expressed in related research literature, on whether "the enemy of my enemy is my friend" [22] or not [23]. It is obvious, from the analysis above, that the transitivity of trust or distrust is safe only in paths that contain at most one negative (distrust) edge. In all other cases, we decide not to propagate trust.

As already mentioned, the weight $cred(u_j \rightarrow u_q, t_c)$ in Equations 6 and 7 is a measure of the credibility of witness u_q, depends on the transitivity horizon considered (i.e., depth in the circle of trust/distrust), and is a function of the local rating attributed to each user in the trust chain.

Let there be P distinct paths of various depths d that connect u_j to u_q through a number of witnesses $u_{q(d)}$ which in line form a trust chain. The weight $cred(u_j \rightarrow u_q, t_c, p)$ for a specific path $p \in P$ of depth $d = n$ is defined as follows:

$$\begin{aligned} cred(u_j \rightarrow u_q, t_c, p) = \frac{1}{n} \cdot & cred(u_j \rightarrow u_q(1), t_c) \\ & \cdot cred(u_q(1) \rightarrow u_q(2), t_c) \cdot ... \\ & \cdot cred(u_q(n-1) \rightarrow u_q(n), t_c) \end{aligned} \quad (8)$$

where $u_q(d)$ denotes the witnesses u_q in p examined at depth d, and analogously to Equation 7,

$$cred(u_q(d) \rightarrow u_q(d+1), t_c) = \frac{LocalRating(u_q(d) \rightarrow u_q(d+1), t_c)}{\sum_{u_{q(d+1)} \in \{W(u_q(d)) \cup u_q(d)\}} |LocalRating(u_q(d) \rightarrow u_q(d+1), t_c)|} \quad (9)$$

where $LocalRating(u_q(d) \rightarrow u_q(d+1), t_c)$ is the local rating attributed to user $u_q(d+1)$ by the evaluator $u_q(d)$. When $d = 0$ the formula calculates the direct reputation weight for the evaluator u_j.

Then, the overall weight $cred(u_j \rightarrow u_q, t_c)$ across all paths $p \in [1..P]$ is defined as the average or the maximum (or maximum) weight across all paths:

$$cred(u_j \rightarrow u_q, t_c) = \frac{\sum_{p=1}^{P} cred(u_j \rightarrow u_q, t_c, p)}{P} \quad (10)$$

or

$$cred(u_j \rightarrow u_q, t_c) = \max_p(cred(u_j \rightarrow u_q, t_c, p)) \quad (11)$$

As may be observed from Equation 8 the transitivity horizon considered is at most n. This is a parameter of the personalized recommendation system in accordance with the specific preferences of the evaluator user. In this work we define the reputation of a witness as a multiplicative function, as shown in Equation 8. Other functions (e.g., minimum of all weights) could be defined. We should note, however, that due to the controversy related with the fourth category of witnesses (enemies of enemies), we assume that this formula only applies to the first three witness categories and only in paths containing at most one negative edge.

D. Trust-Aware Personalized Recommendations

At the end of this process, the model assigns a personalized collaborative reputation rating $CollRating(u_j \rightarrow u_i, t_c)$ for all users u_i who are connected directly or indirectly with the evaluator u_j up to the specific transitivity horizon considered. This rating enables the recommendation model to generate a personalized user ranking for u_j. From this ranking, the top-k users (who are not yet connected to u_j) are provided to the evaluator as positive recommendations (thus, they could be added to the "friend list" $F(u_j)$ of the evaluator user u_j), while the bottom-k users are provided as negative recommendations (thus, they could be added to the "enemy list" $E(u_j)$ of the evaluator user).

V. Experimental Evaluation

In this section we experimentally evaluate our recommender system. It has been proven very difficult to find a social network dataset that combines implicit and explicit trust statements, time information and both positive and negative connections. Similarly, it has been difficult to find a dataset for testing the ability of our recommender in making proper "friends" and "enemies" suggestions to the

users. For the experimental evaluation, we used datasets referenced in the bibliography focusing on those that covered most of the desired characteristics of a social network, as described above. In Section V-A we present results on the *extended Epinions dataset*[1]. This dataset contains both explicit and implicit trust statements between users. In Section V-B we evaluate the ability of our system in recommending trustful connections to the network members using explicit User-to-User connections only. For this purpose we employ the *Advogato dataset*[2], which contains explicit trust statements between users of the $Advogato$ community. Finally in Section V-C we evaluate the performance of our model in predicting positive or negative edges in trust networks with different characteristics and compare with state-of-the-art algorithms in the extended $Epinions$ and $Wikipedia$ vote network[3] datasets.

A. Experiments on Epinions

$Epinions$ is a large product review community that supports various types of interactions between users, such as explicit User-to-User trust statements and product reviews written by the community members and rated by other members. The dataset that we used contains information about product reviews written by the members of the $Epinions$ community. It contains approximately 132,000 members (95,318 after removing self-references) who have issued 841,372 explicit User-to-User statements (85% of them are positive) for 95,318 users and 13,6 million explicit User-to-Item statements for 755,760 different items. More specifically, it contains user ratings that denote which users are trusted or distrusted (1 and -1 respectively) by which users, as well as ratings for product reviews (ranging from 1 to 6). User ratings are the explicit User-to-User connections of our model and review ratings are the explicit User-to-Item connections, which in our experiments carry a positive recommendation meaning (a value of 6 denotes a strong recommendation, whereas a value of 1 denotes a weak recommendation). The dataset also provides the timestamp of each explicit User-to-User trust statement. Finally, the dataset contains information about the author and subject of each review, giving us evidence on each author's interests.

To evaluate our recommendations, we measure the average similarity between a user's interests and those of users in the top-k (i.e., friend) or bottom-k (i.e., enemy) positions in the recommendation list produced by our reputation model. According to Shani and Gunawardana [39], it is unclear how to measure trust in an offline experiment, since trust is build through an interaction between the user and the system. However, according to the same work, it may be beneficial for the system to recommend a few items that the user already knows and likes. In this direction, we capitalize on the similarity of interests between a user and the users recommended by our model and use cosine similarity which is widely used in collaborative filtering to measure the similarity of interests between users [40], [41]. Similarity of users' interests is measured on the corresponding article rating vectors.

In the first set of experiments on the $Epinions$ dataset we examine all users in the dataset. First, we process the complete graph, containing trust and distrust User-to-User statements and all implicit connections that emerge from article ratings (setG: all network members). We evaluate the top-k (i.e., friend) user recommendations (Figure 2) and bottom-k (i.e., enemy) user recommendations (Figure 3), with k ranging from 3 to 30. Then we use all user nodes but only trust statements and article ratings (setE: all members that add positive edges to the network) and evaluate the top-k user recommendations (Figure 4). We also evaluate the bottom-k user recommendations (Figure 5), when all user nodes but only distrust user statements and article ratings are used (setF: all members that add negative edges to the network).

For each user u_j, we compare the lists of recommended users created using the local and the collaborative rating formation and compare against the existing friend list $F(u_j)$ (direct friend list - DFL) or enemy list $E(u_j)$ (direct enemy list -DEL) for each user. In the case of the local rating score, the explicit or implicit positive trust statements of user u_j push the respective users to the top of the friend list and the negative statements push the respective users to the top of the enemy list. In the case of the collaborative rating formation, we use a two-step transitivity horizon, which means that for positive recommendations we aggregate information on the friends of u_j (members in $F(u_j)$) and on their friends, whereas for negative recommendations, we examine the enemies of u_j (members in $E(u_j)$), the enemies of users in $F(u_j)$ and the friends of users in $E(u_j)$. In all the experiments, we set the system memory r to infinity so that all ratings (User-to-User or User-to-Item) at all time periods will be employed.

As explained in Section IV, the local and the collaborative rating formations take into account the direct User-to-User statements. As a result, users in the original DFL (or DEL) lists have a great chance to appear in the top (or bottom) places of the local or collaborative rating lists. Recommending users that are already in the direct friend (or enemy) list is meaningless. So, before evaluating the top-k or bottom-k lists we remove the direct friend or enemies from the corresponding list. The task of making recommendations is now harder, since we must recommend "new" friends, who are not in the direct friend list yet ideally will be more promising friends than the actual members of the DFL.

The average similarity of interests between a user u_i (in setG) and users in his friend or enemy recommendation list are depicted in Figures 2 and 3 respectively. Results show that the average similarity is independent of k, which is reasonable since all friends (or enemies) in $Epinions$ get the same trust (or distrust) score +1 (or -1). The performance of the local friend list (LFL) formation based on the local reputation rating (cf. Equation 4) is worse than that

[1] http://www.trustlet.org/wiki/Extended_Epinions_dataset
[2] http://www.advogato.org
[3] http://snap.stanford.edu/data/wiki-Vote.html

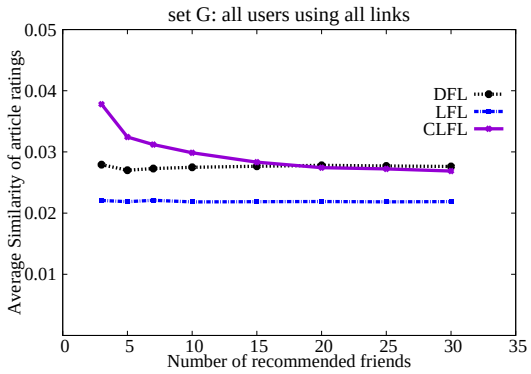

Fig. 2. Similarity between a user and the top-k recommended users (friends), for all users using all links

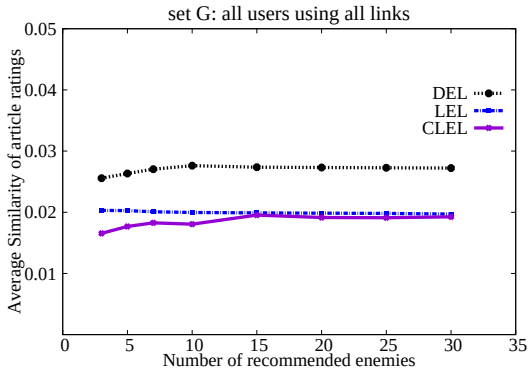

Fig. 3. Similarity between a user and the bottom-k recommended users (enemies), for all users using all links

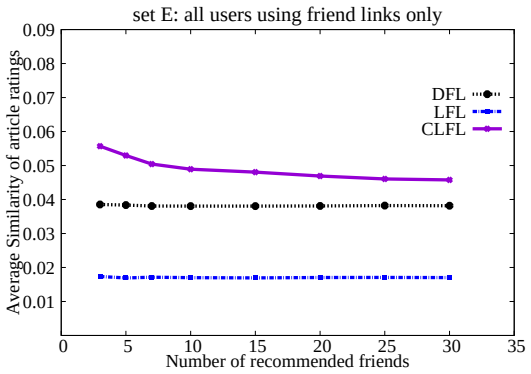

Fig. 4. Similarity between a user and the top-k recommended users (friends), for all users using trust links only

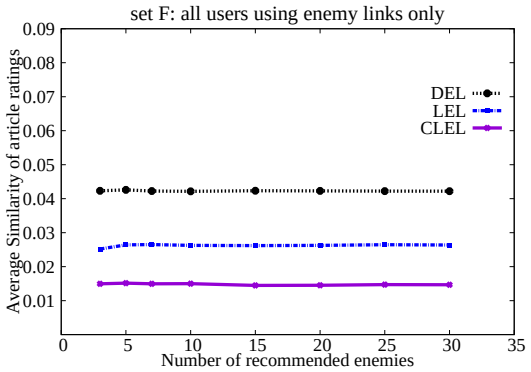

Fig. 5. Similarity between a user and the bottom-k recommended users (enemies), for all users using distrust links only

of DFL and explains our initial thought that recommending "new" friends who are not in the direct friend list is a hard task. The performance of the collaborative local friend list (CLFL) formation based on the collaborative rating (cf. Equation 6) is quite promising, especially when less than the top 10 friend recommendations are evaluated. Results in Figure 3 show that the local enemy list (LEL) that is based on the local rating formation (cf. Equation 4) and the collaborative local enemy list (CLEL) that is based on the collaborative rating formation (cf. Equation 6) outperform DEL (the average similarity between a user and the top direct enemies is higher than that between the user and the recommended enemies). This indicates that both our methods recommend as enemies users that strongly differ in interests from the target user. All the differences depicted in Figures 2 and 3 are statistically significant since the average similarity has been calculated for all the 95,318 users. The average similarity values are small and this is mainly due to the size of the respective vectors, which can be huge but sparse (in *Epinions* users provide article ratings for almost 755,000 different articles). Finally, as expected, the similarity between a user and the recommended friends is bigger than that between the user and his recommended enemies.

In order to study the effect of trust link polarity in the quality of recommendations, we examine the *Epinions* graph using separately positive (Figure 4) and negative (Figure 5) trust statements. This results in a subset of the original user set (setE) comprising 88180 users, which are connected with positive trust links and another subset (setF) comprising 18499 users connected with negative trust links only. We observe that the local rating formation is not sufficient to provide good friend recommendations, but its performance in providing enemy recommendations is acceptable. On the other hand, the improvement in the performance of the collaborative rating formation for both enemy and friend recommendations is better even for higher values of k.

In order to better understand when the two models are able to provide good positive or negative recommendations, we run a second set of experiments on subsets of the *Epinions* dataset. The subsets contain: a) 5057 members with 5–10 friends (setA), b) 4927 members with more than 30 friends (setB), c) 778 members with 5–10 enemies (setC), d) 731 members with more than 30 enemies (setD).

As far as the "friend list" is concerned, the average similarity decreases for big values of k, since less relevant users are added to a long list. This happens mainly with the collaborative rating metric (setA CLFL) and less with the local one (setA LFL), however, CLFL outperforms both LFL and DFL (Figure 6). This proves the ability of the collaborative mechanism to find users of trust in the extended neighborhood of a user and enriching his/her circle of friends. For users with many direct friends (SetB), CLFL still outperforms the direct friend list (DFL) and provides better recommendations than LFL (Figure 7). A

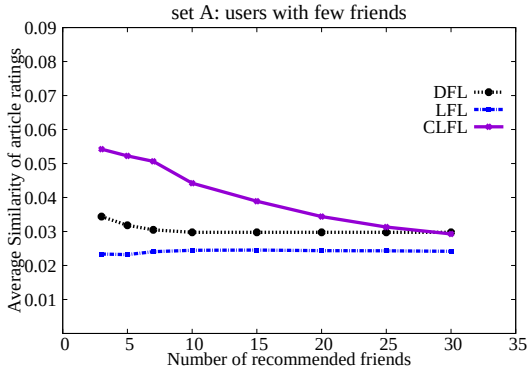

Fig. 6. Similarity between a user and his friend recommendations, for users with few friends

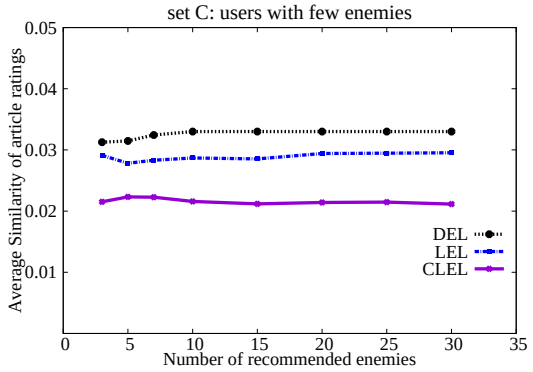

Fig. 8. Similarity between a user and his enemy recommendations, for users with few enemies

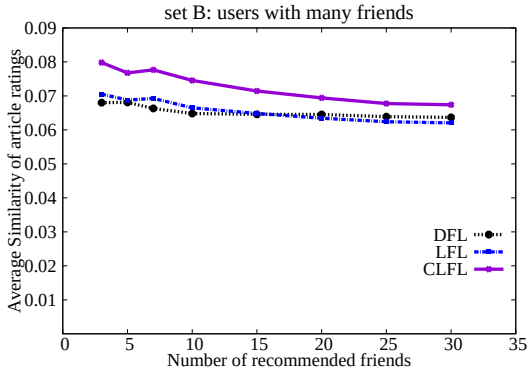

Fig. 7. Similarity between a user and his friend recommendations, for users with many friends

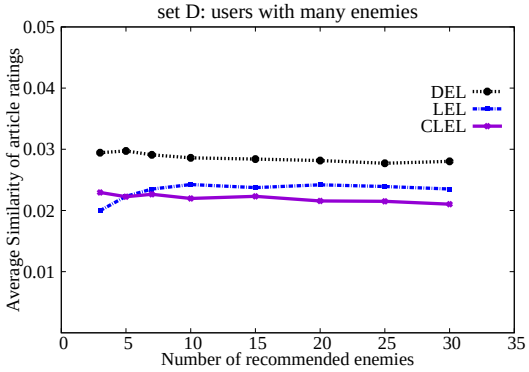

Fig. 9. Similarity between a user and his enemy recommendations, for users with many enemies

reason for this is that long lists of friends result in an overall decrease to the similarity between their interests and those of the user. Thus, members with many friends can benefit from our system, since they can distill their existing friends and find additional friends of high interest to them as suggested by the recommender system.

In the case of "enemy lists", the similarity between the user and the recommended enemies decreases when compared to the direct enemy list. As shown in Figure 8, for users with few direct enemies (setC) the enemy recommendation list based on local rating (LEL) has a higher average similarity than the respective list that is based on the collaborative local rating (CLEL). Both LEL and CLEL achieve average similarity in article ratings between the evaluator and the recommended users less than DEL. For users with many enemies (setD) (Figure 9) the average similarity in article ratings between the user and the recommended users (using either LEL or CLEL) is smaller than that between the user and his/her direct enemies (DEL). This shows that our system recommends as enemies users with few similarities (in article ratings) to the user. For users with a long enemy list, the system can provide recommendations that will further distill this list.

In order to measure the effect of the time decay factor on the quality of recommendations, we repeat the whole set of experiments in sets A to G, this time ignoring the time information. Table I presents the difference between the average similarity values with and without the time

TABLE I
THE EFFECT OF IGNORING TIME INFORMATION ON THE AVERAGE SIMILARITY OF USER RATINGS

SET	LFL	CLFL
A	-0,000105	-0,003550
B	-0,009291	-0,010239
E	-0,000005	0,001188
G_{top}	-0,000225	0,000690
	LEL	CLEL
C	0,000071	0,000175
D	0,002621	0,004969
F	0,000449	0,000452
G_{bottom}	0,000011	-0,000222

decay factor. The difference is averaged on all the top-k cases examined for each dataset. The results in the case of friend recommendations (i.e., sets A, B, E, G_{top}) show that the average performance of LFL always decreases when time decay is ignored, whereas the performance of $CLFL$ decreases for sets A and B. In these sets we consider positive edges only so an interpretation of the above results can be that in networks with many positive trust statements it is important to consider the freshness of these statements in order to provide better friend recommendations. In the case of enemy recommendations (i.e., sets C, D, F, G_{bottom}) results in almost all cases demonstrate a decrease in performance when time decay is ignored (the average similarity scores are higher than in the case of using time decay). The decrease is maximum for setD, where we consider only

negative edges and densely interconnected users.

B. Experiments on Advogato

In order to evaluate the ability of our reputation management model to predict users' reputation we employ the $Advogato$ dataset. $Advogato$ is an online community for software developers created in 1999. $Advogato$ users can certify each other at four levels: Observer, Apprentice, Journeyer or Master. This corresponds to the explicit User-to-User statements of our model. In the absence of User-to-Item information in the $Advogato$ dataset, our model exploits only the explicit User-to-User information. We distribute the four nominal values equally to the [0,1] range (Observer=0.25, Apprentice=0.5, Journeyer=0.75, Master=1) in order to import them to our model. Although we do not have negative trust scores in this case, the task of predicting the correct trust level is not a binary problem (i.e., positive or negative), which further increases its difficulty.

We compare our collaborative rating model against several other trust metrics (both local and global ones) using the "leave-one-out" cross-validation technique as follows: we remove only one trust edge (e.g., from u_j to u_i) from the graph and then we use our reputation model and the remaining graph in order to predict the value of the removed edge. Among the different cross-validation techniques, we choose "leave-one-out", since it has the minimum possible effect on the graph structure (only one edge is removed each time). This is important, since our model employs the whole graph in order to compute trust scores.

The collaborative rating model is evaluated with two different transitivity horizon values, namely: a) transitivity horizon 2 ($CL2$), which means that the evaluator considers the statements of the people he/she trusts and b) transitivity horizon 3 ($CL3$), which means that the evaluator also considers the statements of the people trusted by the people he/she trusts. We evaluate two alternatives of our method: one that takes the average trust score when multiple trust paths exist that connect u_j to u_i, which is called $CLavg$ (as shown in Equation 13) and one that considers the maximum trust score over any of the paths, which is called $CLmax$ (Equation 14). This results in four combinations of transitivity horizon and path selection method, namely $CL2avg$, $CL2max$, $CL3avg$ and $CL3max$.

Using the same evaluation methodology followed in [42], we compare our collaborative rating method with some baseline methods: Random (i.e., predict a random trust score in the range [0, 1]), AlwaysMaster, AlwaysJourneyer, AlwaysApprentice, AlwaysObserver (i.e., always predict a Master, Journeyer, score etc.), Outu_j (i.e., the trust that u_j assigns to any other user u_i is always the average trust score assigned by u_j), Inu_i (i.e., the trust assigned to a user u_i by any user u_j equals to the average trust score assigned to u_i by the users that trust u_i). We also compare against PageRank, but first sort and rescale (linearly map) PageRank values in the range [0, 1]. Additionally, we compare against a well-known referral-based propagation approach $TidalTrust$[4] [43] and the local version of the $Advogato$ metric[5]. Finally, we compare against a recently proposed metric called $Shin$ [44], which takes into account trust propagated through reachable witnesses and trust estimated in unreachable witnesses based on common acquaintances. In our implementation, we assume a depth of 3 and propagate trust through all reachable witnesses, using $CertProp$ as suggested in [44] with γ = 100%. Additionally, we estimate trust for all the unreachable witnesses and keep the path that gives the highest trust score.

TABLE II
RESULTS ON ADVOGATO DATASET

		Evaluation Metrics				
		MAE	Recall	Precision	F_1	F_{1bal}
Baseline	AlwMaster	0.253	1.000	0.899	0.947	0.667
	AlwJourneyer	0.168	1.000	0.899	0.947	0.667
	AlwApprentice	0.290	1.000	0.899	0.947	0.667
	AlwObserver	0.497	0.100	1.000	0.182	0.667
	Random	0.364	0.503	0.914	0.649	0.508
SoA methods	Outu_j	0.181	0.954	0.922	0.938	0.680
	Inu_i	**0.126**	0.959	0.921	0.940	0.683
	PageRank	0.368	0.917	0.920	0.918	0.657
	TidalTrust3 [43]	0.132	0.946	0.918	0.932	0.694
	TidalTrust4 [43]	0.128	0.942	0.922	0.932	0.693
	Shin [44]	0.128	0.932	0.938	0.935	0.736
	Advogato [45]	0.245	**0.979**	0.942	**0.960**	**0.752**
Collaborative	CL2avg	0.148	0.889	0.974	0.930	0.728
	CL2max	0.133	0.937	0.947	0.942	0.738
	CL3avg	0.174	0.820	**0.980**	0.893	0.710
	CL3max	0.149	0.963	0.934	0.948	0.739

The predicted values are either compared to the real values or are mapped to a binary problem and evaluated using: a) the mean absolute error ($MAE = \sum_{i=1}^{n} |predictedTrust_i - actualTrust_i|/n$, for n edges), which averages the absolute difference between the real and predicted values b) recall, c) precision and d) F_1 score. The mean absolute error is applied on the exact values predicted by each model, whereas for recall, precision and F_1, we examine the problem as a binary classification problem (i.e., a trust score $\geq$ 0.5 is a positive and a trust score <0.5 is a negative example). Comparisons are repeated for all the examined edges and the average values are depicted in Table II.

The results in the first zone of Table II (Baseline methods) are strongly related to the distribution of edges' values in the $Advogato$ dataset. We can see that Journeyer is the most common edge value and as a consequence, a trust metric that always predicts this value has better chances than the other three metrics (i.e., AlwMaster, AlwApprentice and AlwObserver) and of course better than the random prediction. When we examine the binary classification problem, the first three edge types map to the same class (i.e., edge) and significantly outnumber the

[4]We employed the Java implementation provided here: http://code.google.com/p/happy-coding-projects/

[5]We employed the Java implementation of $Advogato$ trust metric provided here: http://ftp.saddi.com/pub/software/advogato-tmetric-asaddi-0.2.tar.gz

Observer type (i.e., no-edge). As a result, we have high chances to predict accurately when we always predict an edge in this leave-one-out experiment. From the 57,568 trust edges contained in the dataset, only 10% correspond to negative (i.e., Observer) edges, which results in a skew of the results (precision values are always greater than 0.9 and recall values greater than 0.1) and favors metrics that always suggest high trust scores (e.g., AlwMaster). In order to avoid this skew, we repeated the experiment examining an equal number of positive and negative examples (5,000 Observers and 5,000 from the other three levels). The last column in Table II (F_{1bal}) presents the F_1 scores in this case.

The second zone of Table II contains the results of several state of the art (SoA) methods. As far as the Outu_j and Inu_i metrics are concerned, they can be considered complementary, since they average the values of the outgoing trust connections of u_j and the incoming trust connections to u_i respectively, in order to predict a value for the edge from u_j to u_i. The Outu_j metric is very fast and processes only information available to user u_j. However, it assumes that u_j assigns the same score for every user in the network. Similarly, Inu_i always predicts the same score for u_i independently of the evaluator user. Inu_i is more democratic in nature and usually more reliable than Outu_j. The equivalents of Inu_i and Outu_j in the Web are hub and $authority$ scores respectively. The main drawback of Inu_i is that it requires knowledge of all the trust ratings assigned to u_i by other users in the network, which in the case of a distributed network may be very slow or even infeasible. The $PageRank$ metric provides a global score for the members of the network, taking into account all the trust connections of the network. It is the slowest of all metrics, even in the centralized case, where all ratings are available and stored in the same repository. The $Advogato$ metric does not require knowledge of the whole network, since it follows a spreading of activation technique in a portion of the graph that contains the evaluator user u_j and the target u_i. However, it is slower than our collaborative rating metrics, at least in the implementation we employed, and its MAE is worse than all our metrics, which means that it is worst in predicting the exact value of a trust link. Similarly, $TidalTrust$ visits part of the network each time, since it starts from the source of the examined edge and follows a breadth-first search of the network in a limited horizon (e.g. depth of 3 or 4 in our experiments). Results show that $TidalTrust$ has one of the lowest MAE but its F_1 scores are worse than all other local metrics and our Collaborative Local metrics that use the path of maximum trust each time. The implementation for $Shin$ can be very fast (e.g. for depth 3 that we experimented) when the witnesses of each node are known in advance (backtrack trust links), which, however, assumes that the whole graph is known in advance. Its results are good, although worse than those of $Advogato$ metric and of most of the Collaborative Local metrics presented in the third zone of Table II (Collaborative methods).

The four versions of our collaborative rating metric ($CL2avg$, $CL2max$, $CL3avg$, $CL3max$) differ in the score they assign to the edge from u_j to u_i when there exist more than one paths that can be employed for the prediction and in the horizon of the transitivity of trust. CL2avg and CL3avg take the average score for all paths, whereas CL2max and CL3max take the maximum score, which correspond to trusting the path with the most trustworthy nodes. Comparing between average and maximum values, we see that when multiple paths exist between the evaluator and the target user in the $Advogato$ dataset, it is better to consider the path with the maximum value. This is reasonable, since it is based on the most trustworthy path of witnesses, but cannot be generalized in all networks, especially in networks with few trustworthy and many untrustworthy edges.

According to the results presented in Table II the Inu_i metric is better than any other metric when the mean average error (MAE) is considered. However Inu_i (and similarly $PageRank$) is a global metric. This makes its implementation in a distributed environment or in very large networks infeasible, since it requires incoming link knowledge, which is not directly (or indirectly) available to u_i. Our collaborative local metrics ($CL2max$ and $CL2avg$) provide the second and third best results (in MAE) and are better than the local metric of $Advogato$, with $CL2max$ having a slight advantage in performance over CL2avg. The lower performance of CL3 metrics, when compared to their CL2 equivalents, can be due to the arbitrary quantification of nominal trust statements (Master, Journeyer, Apprentice, Observer) to numerical values (1, 0.75, 0.5, 0.25). However, in the binary classification problem $CL3avg$ demonstrates the highest $Precision$ score among all other methods and clearly outperforms $Advogato$ and $PageRank$. When an equal number of positive and negative examples is employed as shown in the F_{1bal} column, then our metrics outperform all other metrics, except $Advogato$. Once again, the results show that information from the circle of trust can assist in predicting trust connections and may provide useful user recommendations to the network members.

Summary of results. Our findings on the $Advogato$ dataset can be summarized as follows:

- The Inu_i metric has the lowest error in predicting trust scores. However its main disadvantage is that it is a global metric, i.e., requires knowledge of all trust ratings assigned to a user. This is not always possible to implement in a distributed environment (e.g., on a mobile social networking application that stores trust data on clients) or in very large networks.
- The PageRank metric is global too, but its performance is worse than most of the metrics. PageRank seems to be an improper solution for trust networks, because trust propagation has a limited horizon.
- Local metrics that propagate trust in a limited horizon, such as $Advogato$, $TidalTrust$, $Shin$ and CL have a better performance than PageRank and are usually faster than PageRank, since they do not examine the whole graph and they do not have iterations.

TABLE III
DATASET STATISTICS

	Epinions	Wikipedia
Nodes	119,217	7,118
Edges	841,200	103,747
+edges	85.0%	78.7%
-edges	15.0%	21.2%

TABLE IV
PREDICTIVE ACCURACY FOR POSITIVE AND NEGATIVE EDGES IN DIFFERENT DATASETS

	Epinions			Wikipedia		
Method	Acc	Pred+	Pred-	Acc	Pred+	Pred-
BalanceLrn	0.902	-	-	0.756	-	-
BalanceDet	0.771	-	-	0.578	-	-
WeakBalance	0.823	-	-	0.608	-	-
StatusLrn	0.831	-	-	0.716	-	-
StatusDet	0.808	-	-	0.668	-	-
CL2avg	0.957	0.73	0.48	0.861	0.76	0.44
CL3avg	0.924	0.90	0.82	0.857	0.89	0.62

- The performance of our metrics is comparable to state of the art local metrics, such as $Advogato$, $TidalTrust$ or $Shin$ when a binary classification is assumed for an edge (trust or not-trust). $Advogato$ is slower and its error in trust score prediction is worse than ours. $Shin$ is faster since it achieves higher coverage in the same depth as other metrics, since it assumes trust for unreachable witnesses.
- When multiple paths exist between two users, considering the path with the maximum value (e.g. in CLmax) gives better predictions.

C. Generalization across datasets

In this section, we evaluate the generalization of our model and its applicability in trust networks with different topologies and trust semantics. In this set of experiments we compare our system with the most related state-of-the-art work of Leskovec *et al.* [23]. We apply our model on two datasets, the extended Epinions and the Wikipedia vote network, following the same leave-one-out cross validation methodology. We try to predict both positive and negative edges, which in our model may result in a positive, negative or zero score. Since in some cases the edge is not predicted at all from our model, we give evidence on the coverage of our model in the case of positive and negative edges. The statistics of the two datasets are reported in Table III.

Table IV shows the results of our experiments (using $CL2avg$ and $CL3avg$ as in section V-B), along with the best results presented in [23].The comparison shows that both $CL2avg$ and $CL3avg$ outperform the best methods reported in [23]. The accuracy of $CL2avg$ is higher, however its ability in predicting an edge, either positive or negative, is worse than that of $CL3avg$.

VI. CONCLUSIONS

This work presents a trust-aware system for generating personalized user recommendations in social networks. Its foundations lie on a reputation mechanism which is mathematically formulated, comprising both local and collaborative rating formation. Our system exploits two special features of social networks: a) the difference between explicit trust statements which carry stronger trust semantics and implicit trust statements, which represent a more transient reference to another network member, b) the timestamp information of a connection between users, either implicit or explicit. Moreover, the model is able to handle negative trust (distrust) statements and supports transitivity of trust under conditions.

Our initial experiments in three real-life datasets show that the designed framework performs well. Specifically, our model outperformed other local metrics achieving higher precision and recall when a binary classification is considered (trust/no-trust) and lower mean average error when real trust values are predicted. Additionally, the collaborative rating metric performs better than the local one. For users with few connections, the recommender system suggests new users of high interest, whereas for users that already have long lists of friends or enemies, the system can provide recommendations that will help them to further distill these lists.

Our future plans include the study of more social network datasets, by applying our trust-aware system to them and evaluating its performance in different setups: with and without timestamp information, with and without explicit and implicit connections, with and without negative connections and transitivity of trust. Finally, we intend to apply and evaluate our recommendation system on a social network in a real-time scenario.

REFERENCES

[1] S. Perugini, M. Goncalves, and E. Fox, "Recommender systems research a connection-centric survey," *Journal of Intelligent Information Systems*, vol. 23, no. 2, pp. 107–143, 2004.

[2] B. Yang, W. Cheung, and J. Liu, "Community mining from signed social networks," *IEEE Trans. on Knowl. and Data Eng.*, vol. 19, no. 10, pp. 1333–1348, 2007.

[3] I. Guy, N. Zwerdling, D. Carmel, I. Ronen, E. Uziel, S. Yogev, and S. Ofek-Koifman, "Personalized recommendation of social software items based on social relations," in *RecSys*, 2009, pp. 53–60.

[4] I. Konstas, V. Stathopoulos, and J. M. Jose, "On social networks and collaborative recommendation," in *SIGIR*, 2009, pp. 195–202.

[5] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization," in *CIKM*, 2008, pp. 931–940.

[6] S. Nakajima, J. Tatemura, Y. Hino, Y. Hara, and K. Tanaka, "Discovering important bloggers based on analyzing blog threads," in *2nd Annual Workshop on the Blogging Ecosystem: Aggregation, Analysis and Dynamics*, 2005.

[7] A. Kritikopoulos, M. Sideri, and I. Varlamis, "Blogrank: Ranking on the blogosphere," in *ICWSM*, 2007.

[8] E. Adar, L. Zhang, L. Adamic, and R. Lukose, "Implicit structure and the dynamics of blogspace," in *Workshop on the Blogging Ecosystem, WWW*, 2004.

[9] I. Varlamis and M. Louta, "Towards a personalized blog site recommendation system: A collaborative rating approach," in *4th Intl. Workshop on Semantic Media Adaptation and Personalization (SMAP)*, 2009.

[10] M. D. Louta and I. Varlamis, "Blog rating as an iterative collaborative process," in *Semantics in Adaptive and Personalized Services*, ser. Studies in Computational Intelligence, M. Wallace, I. Anagnostopoulos, P. Mylonas, and M. Bieliková, Eds. Springer, 2010, vol. 279, pp. 187–203.

[11] P. Massa and P. Avesani, "Controversial users demand local trust metrics: An experimental study on epinions.com community," in *AAAI*, 2005, pp. 121–126.

[12] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "Twitterrank: finding topic-sensitive influential twitterers," in *WSDM*, 2010, pp. 261–270.

[13] J. Golbeck, "Trust and nuanced profile similarity in online social networks," *TWEB*, vol. 3, no. 4, 2009.

[14] P. Massa and P. Avesani, "Trust metrics in recommender systems," in *Computing with Social Trust*, J. Golbeck, Ed. Springer London, 2009, ch. 10.

[15] F. E. Walter, S. Battiston, and F. Schweitzer, "Personalised and dynamic trust in social networks," in *RecSys*, 2009, pp. 197–204.

[16] V. Cahill, E. Gray, and J.-M. S. et al., "Using trust for secure collaboration in uncertain environments," *IEEE Pervasive Computing*, vol. 2, pp. 52–61, 2003.

[17] S. I. Ahamed, M. M. Haque, and N. Talukder, "A formal context specific trust model (ftm) for multimedia and ubiquitous computing environment," *Telecommunication Systems*, vol. 44, no. 3-4, pp. 221–240, 2010.

[18] T. Grandison and M. Sloman, "A survey of trust in internet applications," *IEEE Communications Surveys and Tutorials*, vol. 3, no. 4, pp. 2–16, 2000.

[19] J. Chen, W. Geyer, C. Dugan, M. J. Muller, and I. Guy, "Make new friends, but keep the old: recommending people on social networking sites," in *CHI*, 2009, pp. 201–210.

[20] I. Guy, M. Jacovi, E. Shahar, N. Meshulam, V. Soroka, and S. Farrell, "Harvesting with sonar: the value of aggregating social network information," in *CHI*, 2008, pp. 1017–1026.

[21] I. Guy, I. Ronen, and E. Wilcox, "Do you know?: recommending people to invite into your social network," in *IUI*, 2009, pp. 77–86.

[22] J. Kunegis, A. Lommatzsch, and C. Bauckhage, "The slashdot zoo: mining a social network with negative edges," in *WWW*, 2009, pp. 741–750.

[23] J. Leskovec, D. P. Huttenlocher, and J. M. Kleinberg, "Predicting positive and negative links in online social networks," in *WWW*, 2010, pp. 641–650.

[24] I. Varlamis, M. Eirinaki, and M. D. Louta, "A study on social network metrics and their application in trust networks," in *ASONAM*, 2010, pp. 168–175.

[25] ——, "Application of social network metrics to a trust-aware collaborative model for generating personalized user recommendations," in *The Influence of Technology on Social Network Analysis and Mining*, ser. Lecture Notes in Social Networks, T. Özyer, J. G. Rokne, G. Wagner, and A. H. P. Reuser, Eds. Springer, 2013, vol. 6, pp. 49–74.

[26] J. Golbeck, "Trust on the world wide web: a survey," *Found. Trends Web Sci.*, vol. 1, no. 2, pp. 131–197, 2006.

[27] D. Cartwright and F. Harary, "Structural balance: a generalization of heider's theory," *Psychological Review*, vol. 63, 1956.

[28] J. O'Donovan, "Capturing trust in social web applications," in *Computing with Social Trust*, J. Golbeck, Ed. Springer, 2009, ch. 9.

[29] J. Golbeck, "Generating predictive movie recommendations from trust in social networks," in *iTrust*, 2006, pp. 93–104.

[30] R. V. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust," in *WWW*, 2004, pp. 403–412.

[31] P. Massa and P. Avesani, "Trust-aware recommender systems," in *RecSys*, 2007, pp. 17–24.

[32] C.-N. Ziegler and J. Golbeck, "Investigating interactions of trust and interest similarity," *Decision Support Systems*, vol. 43, no. 2, pp. 460–475, 2007.

[33] C.-N. Ziegler, "On propagating interpersonal trust in social networks," in *Computing with Social Trust*, J. Golbeck, Ed. Springer London, 2009, ch. 6.

[34] F. E. Walter, S. Battiston, and F. Schweitzer, "A model of a trust-based recommendation system on a social network," *Autonomous Agents and Multi-Agent Systems*, vol. 16, no. 1, pp. 57–74, 2008.

[35] D. Liben-Nowell and J. M. Kleinberg, "The link prediction problem for social networks," in *CIKM*, 2003, pp. 556–559.

[36] C. de Kerchove and P. V. Dooren, "The pagetrust algorithm: How to rank web pages when negative links are allowed?" in *SDM*, 2008, pp. 346–352.

[37] A. Kale, A. Karandikar, P. Kolari, A. Java, T. Finin, and A. Joshi, "Modeling trust and influence in the blogosphere using link polarity," in *ICWSM*, 2007.

[38] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *WWW*, 2003, pp. 640–651.

[39] G. Shani and A. Gunawardana, "Evaluating recommendation systems," in *Recommender Systems Handbook*, 2011, pp. 257–297.

[40] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *SIGIR*, 1999, pp. 230–237.

[41] A. Gunawardana and G. Shani, "A survey of accuracy evaluation metrics of recommendation tasks," *J. Mach. Learn. Res.*, vol. 10, pp. 2935–2962, December 2009.

[42] P. Massa, K. Souren, M. Salvetti, and D. Tomasoni, "Trustlet, open research on trust metrics," *Scalable Computing: Practice and Experience, Scientific International Journal for Parallel and Distributed Computing*, vol. 9, no. 4, pp. 341–351, 2008.

[43] J. Golbeck, "Computing and applying trust in web-based social networks," *PhD Dissertation*, 2005.

[44] C.-W. Hang, Z. Zhang, and M. Singh, "Generalized trust propagation with limited evidence," *Computer (to appear)*, 2013.

[45] R. Levien, "Advogato's trust metric." [Online]. Available: http://www.advogato.org/trust-metric.html

Magdalini Eirinaki is an Assistant Professor at the Computer Engineering Department, San Jose State University, California. Her research interests cover the areas of web mining and recommendation systems and, in particular, on personalization, interactive database exploration and mining of social networks. She has published several papers in refereed journals and international conference proceedings in the above areas. She received her Ph.D. degree in Computer Science from Athens University of Economics and Business, Greece in 2006.

Malamati D. Louta is Assistant Professor at the Department of Informatics and Telecommunications Engineering, University of Western Macedonia, Greece. She holds M.Eng. (1997) and Ph.D. (2000) degrees in Electrical and Computer Engineering and M.B.A. degree (2004) from the National Technical University of Athens. Her research interests include telecommunication networks and advanced services engineering. She is the author of over 60 peer-reviewed publications. Dr Louta serves as an associate editor, as a technical program committee member and as a reviewer in a number of international conferences and journals. She is member of the IEEE, the ACM and the Technical Chamber of Greece.

Iraklis Varlamis is a Lecturer at the Department of Informatics and Telematics of Harokopio University of Athens. He received his Ph.D. in Computer Science from Athens University of Economics and Business, Greece. His research interests vary from data mining and the use of semantics in web mining to virtual communities and their applications. He has published several articles in international journals and conferences, concerning web document clustering, the use of semantics in web link analysis and web usage mining, word sense disambiguation using thesauruses, etc.