@2017 Oxford University Press. This is the post-print version of the paper whose published version is available at DOI: https://doi.org/10.1093/comjnl/bxx072

Using Local Trust for Forming Cohesive Social Structures in Virtual Communities

Lidia Fotia[1], Fabrizio Messina[2], Domenico Rosaci[1] and Giuseppe M.L. Sarné[3]

[1]*Department DIIES, University "Mediterranea" of Reggio Calabria, Loc. Feo di Vito, 89122 Reggio Cal. RC*
[2]*Department of Computer Science and Mathematics, University of Catania, Viale A.Doria 6, 95126 Catania CT*
[3]*Department DICEAM, University "Mediterranea" of Reggio Calabria, Loc. Feo di Vito, 89122 Reggio Cal. RC*
Email: messina@dmi.unict.it

Matching users profile in virtual communities can represent the most natural way of representing group homogeneity, i.e. how much the group members are mutually linked. However, optimizing profile matching does not guarantee the group cohesion, i.e. that the group will continue to be homogeneous in time. Moreover, computing profile matching in large virtual communities can be very expensive, and cannot be integrated in a fully distributed system. In the past, we have demonstrated that using users mutual trust, along with profile matching, can help to improve groups homogeneity. In this work we demonstrate, by an extended set of experiments on datasets extracted from real communities, that trust measures can effectively replace profile matching in order to optimize group's cohesion. A further interesting result is represented by the fact that it is also possible to replace the global trust measure with a local measure of trust, called local reputation, which is not highly sensitive to the size of the network, thus allowing to perform computations which are limited on the size of the ego-network of the single node.

Keywords: trust; reputation; virtual communities; group formation; knowledge

1. INTRODUCTION

Nowadays, virtual communities are evolving as complex systems composed by human and software agents. Large social networks, as Facebook [1] and Twitter [2], involve hundreds of millions of users (in 2015, Facebook had 1.6 billions of active users and Twitter surpassed 300 millions of members), and several online communities as, for instance, those based on Internet Relay Chat, include software agents in their members, acting as users' assistants, channel administrators etc. Overall, such communities are organized in *social structures*, where each structure emerges on the basis of some kind of social relationship. The cases of the *facebook friends* and the *twitter followers* are probably the most known examples, but some other types of structures exist, based on other social relationships than the simple "friendship". For example, often the users form "groups "based on some thematic interest, as in the cases of Facebook groups and Twitter lists. In other cases, groups are created for representing classes in e-Learning activities, or set of customers interested in performing together e-Commerce purchases. For instance, a large organization with geographically dispersed offices can use Elgg[4] on an internal server to introduce employees to each other and to share internal knowledge across offices; moreover, employees of firms can use groups to create online study communities, and no-profit organizations can build communities of common interest where members learn from each other. Furthermore, some distributed systems as, for instance, Grid virtual organizations[3] or cloud of clouds[4], can be viewed as virtual communities whose members are software agents that perform activities implying "social "interactions like negotiations, collaborations or competitions.

The formation of a group is often a process that is activated by individual initiative and spontaneously evolves in time, by the continuous repetition of two main events: (*i*) a user of the community asks for joining with a group in which she/he is interested; (*ii*) a group, by means of its administrators, accepts or

[4]https://elgg.org/

refuse the request of an aspirant new member. These two types of events are the consequence of two main activities that are performed in a virtual community, namely that of a user looking for interesting groups and that of a group administrator having to manage her/his group. To be effective, these two activities should produce groups whose members are satisfied to their memberships.

Overall, members of virtual communities would expect that the group satisfies some *requirements* she/he desires as, for example, the topics that are discussed or the level of politeness. As another example, in a cloud system, a software agent could require some given computing resources, and a group of software agents may offer resources satisfying that requirement.

In a past research [5] we have extended the concept of profile similarity, well-known in recommender systems [6], in order to define a measure, called *Average Similarity* AS_g, for representing the global, mutual satisfaction perceived by all the members of a group g of a virtual community. The notion of AS_g takes into account only one requirement for defining the value of satisfaction of a member a with respect to another member b, namely the *similarity* between the profiles of a and b, where the profile of a member contains some information about member's interests and preferences. Therefore, AS_g, for a given group g, was defined as the average satisfaction computed on the totality of couples of members a, b. However the notion of AS_g was based on similarity requirement for the user's satisfaction, it can be easily extended to the general case of multiple requirements, and in this paper we provide such a generalization, that we will call *Average Matching* AM_g.

The *profile matching* between user's requirements and group's characteristics can be considered as the most natural way to represent the *group homogeneity*, measuring how much the group members are mutually linked. However, optimizing profile matching does not guarantee the group *cohesion*, i.e. that the group will continue to be homogeneous in time.

An important issue highlighted in our past proposal [5] was that the satisfaction of an agent a to be in the same group with another agent b should take into account, besides the matching m_{ab} between the characteristics of a and b, also the *trustworthiness* t_{ab} that a has in b. Indeed, an agent, when declaring some characteristics of the profile, could be imprecise or even fraudulent, and this can affect the effectiveness of the matching. Moreover, computing profiles similarities is not always possible, due to the fact that in many cases users do not make publicly available this information.

In our past approach, we have proposed to take into account the trustworthiness in the definition of another measure, called *Average Compactness* AC_g of the group g. In words, AC_g is computed by averaging on all the pair of members a,b of the groups, a measure $C(a, b)$ taking into account both the similarity between a and b and the trustworthiness that a has in b.

On the basis of the compactness measure, we have proposed an algorithm, called User-to-Group (U2G), able to heuristically provide a good solution to the problem of maximizing the Mean Average Compactness (MAS) of all the groups of the social community. We proved the effectiveness of U2G both theoretically and experimentally.

1.1. Research questions and contributions

However, two questions were left open in [5].

1.1.1. A criterion for proving the advantage of the compactness

First, our idea of mixing profile matching and trust measures in a unique measure (compactness) intuitively suggests that groups formed based on compactness, should be capable to exhibit an internal cohesion in time, even in absence of sufficient and sufficiently reliable information about profile matching. Nevertheless we need to experimentally confirm such an intuition, i.e. the effect of considering trust measure, showing that the maximization of the mean average compactness on which trust is weighting more than the profile matching, will produce a good cohesion in real virtual communities. To do this, we need to define a reasonable criterion for measuring the actual capability of the group to not decreasing in time the mutual profile matching of their members.

In this paper, as a first contribution, we propose a criterion to compare, in terms of cohesion, two different configurations S_1 and S_2 of groups for a given virtual community.

To this purpose we introduce a measure called Δ-*Cohesion* $\Psi_\Delta(S)$, defined as follows: the algorithm U2G is applied to the virtual community, starting from the configuration S, in a time-window $[t_0, t_f]$ of largeness equal to $\Delta = t_f - t_0$. The mean average matching MAM, mean of the values AM_g, obtained at the end of the time-window, is called the Δ-*Cohesion* of S in that time-window.

Obviously, a group configuration $S1$ having a $\Psi_\Delta(S1)$ greater than the $\Psi_\Delta(S2)$ of another group configuration $S2$ can be considered more cohesive than $S2$ in the given time-window Δ, since it finally produces groups that in average present better group matching values.

We performed experiments on the real social network CIAO and EPINIONS, obtaining that the use of the sole trust measures produces results, in terms of cohesion, comparable with those obtained considering the profile matching.

1.1.2. Using local reputation rather than reliability

Secondly, in the experiments performed in [5], we have used a measure of the trustworthiness t_{ab} that the agent a has in the agent b, only taking into account the direct knowledge that a has about b. However, in many

cases a does not know b, and thus it is impossible to estimate t_{ab}. As a theoretical proposal, we had also introduced a more general trust measure, by averaging two components rel_{ab} and rep_b, where (i) rel_{ab} is the direct *reliability* of b, i.e. the trustworthiness that a has in b based on the past interactions between a and b, while (ii) rep_b is the global *reputation* of b, i.e. the trustworthiness that all the community has in b. Roughly speaking, in order to compute how much a trusts in b, we have considered the possibility to use both the direct experience of interaction between the two agents, and the experience that the whole set of the agents present in the community had with b. The reason of this choice, very common in many trust-based approaches proposed in the literature, was due to the necessity, when a does not have a sufficient direct knowledge of b, to use the recommendations coming from the other agents of the community. However, as we said above, the reputation does not have been really used in the experiments described in [5], that was limited to the sole reliability. Furthermore, in a next study presented in [7], it has been highlighted that computing the reputation in the way proposed in [5], i.e. taking into account the recommendations provided by *all* the agents of the community, although largely used in past recommender systems, shows a limited effectiveness in estimating trustworthiness of unknown agents, due to the uncertainty about the reliability of the recommendations, especially in those communities having a large number of members. In other words: who does certify the reputation? Moreover, the computation of the global recommendations implies that each agent a has to access to all the global reputation values of all the other agents, making necessary to continuously compute and maintain a central repository for storing this values. This requirement is not suitable for a distributed architecture, as it is often that implementing a virtual community.

In [7], we proposed to integrate the traditional use of the global reputation with another form of reputation, called *local reputation*, that is based on recommendations only coming by the entourage of the user (friends, friends of friends and so on) and thus probably more reliable than completely unreferenced recommendations. The experiments made on real social networks showed that the use of local reputation generally improves the effectiveness of the recommendations with respect to the use of the global reputation and, in most of the cases, the use of the sole local recommendation is the best choice.

In this paper, as a second contribution, we propose to compare the use of the local reputation vs the simple reliability when using the algorithm U2G for automatically forming groups in virtual communities. Our experiments, performed on the real data extracted from virtual communities EPINIONS and CIAO [8], in which users provide reviews concerning commercial products falling in different categories, clearly show that the use of local reputation outperforms the results obtained using the reliability.

It is important to highlight that our approach can be iteratively applied, when profiles and trust values change, for building a new, better solution based on the last founded solution, thus providing a method that incrementally improves its performances without the necessity of starting from scratch data. Moreover, since our U2G algorithm incrementally works, building new solutions based on the previous ones, it is very suitable to be applied in very large networks, having the possibility to continuously improve its performances, adding new samples in time. It is worth to point out that the use of local reputation, that we will show that it is the most effective contribution, is not highly sensitive to the global size of the network, since the local size of each node is limited enough.

2. RELATED WORK

Virtual communities, composed by humans and software agents, represent a relevant area of research, as reported in [9, 10, 11]. An overwhelming corpus of analysis and models on trust has involved fields as computer science [12, 13], sociology [14], economy, etc. In fact, trust affects decisional processes and social interactions, in a plethora of human and virtual activities [15, 16], to improve benefits or mitigate risks for unreliable partners [17]. For example, in [18] trust is used for an agreement procedure used to form groups of trustworthy agents in a P2P system, while in [19] trustworthiness of data is taken into account, and evaluated by data provenance, i.e. the derivation history of data, and authors of [20] take into account the problem of trust inference for the several components of a distributed system.

Several approaches deal with the problem of suggesting items to the members of a group or the groups a user can join with, respectively known as *group recommendation* and *group affiliation* problems. In detail, in order to identify the best items to suggest to a group some approaches adopt a *score aggregation* strategy to build a group profile; it is equivalent to compute a function receiving an item as input and returning how much it satisfies the group members. Here, two popular strategies are the *Average* [21] and the *Least Misery* [22]. Other approaches match users and group information. In particular, in [23], the user that has to trust a stranger, has to form stereotypes by considering features, obtained by the analysis of the profiles and past behaviors with "similar" people, and then aggregates all those matching with the stranger's profile to predict the correspondent value of trust relation.

Recommending new friends to users as well as communities they could join with are generated with a probabilistic approach in [24], where a *bag of users* and a *bag of words* describing a community and its interests

are built and combined in order to improve their data sparsity degree. Differently, in Vasuki *et al.* [25] the co-evolution of the user's friendship relationships and the knowledge of group affiliations are used to predict the next groups to join with.

Trust in social communities is often modeled by a *trust network* (i.e. a graph whose vertexes represent the users and oriented edges represent trust relationships), which is usually sparse. Different techniques use the graph topological properties for inferring new trust values by starting from existing ones. For example, in [26] a maximum network flow algorithm infers trust and in [27] a modified Breadth First Search collects multiple reputation scores, basing on a voting algorithm, and returns a unique reputation rate for each user. Paths up to a fixed length on a trust network are used in [28] to propagate known trust values to get new ones.

To compute trust in virtual communities, the main aspects to consider are: (*i*) informative sources [29]; (*ii*) aggregation rules [30]; (*iii*) trust inference [31]. The first one consists of direct opinions (reliability) based on personal past experiences and/or indirect information (reputation) provided by other users, often aggregated in a unique synthetic trust value [32, 33]. In particular, in large communities each user usually interacts with a narrowest share of his/her community so that reputation is usually predominant on reliability to build a trustworthy opinion about someone.

Moreover, trust can be computed by adopting a local or a global approach in a centralized or distributed way. Some researches states as the local trust is the more accurate when personal users' point of views are adopted [34] with a computational cost depending on the horizon chosen to discovery a *trust* chain linking two users [35]. In Golbeck [36] the shortest paths always provide the most accurate results in inferring trust values, while [31] experimentally compares several strategies to infer trust values by pointing out that the weighted mean aggregation of all paths returns the best accuracy in predicting users' trust rates.

In this context, three known algorithms are TidalTrust [36], MoleTrust [37] and TrustWalker [38]. TidalTrust assumes that the most accurate trust predictions come more from the closer neighbors, although some valuable scores or paths between two users could be ignored if the trust networks are too sparse. MoleTrust [37] computes users' trust values by using a backward exploration and fixing a maximum depth in the search-tree of the trust network where the trust score of a user at depth x is calculated only by using the trust scores computed at depth $x-1$. To suggest items, TrustWalker considers ratings for the target item and for the similar items to solve the lost of precision for data sparsity by using (*i*) a random walk on the trust network and ratings on similar items to limit the depth search on the network and (*ii*) a probabilistic approach to select items by preferring the nearer raters and items similar to the target.

In [39] it is assumed that each link connecting users can represent an interest to a content or the trust into a user where the trust between each pair of users belonging to the same neighbor is explicitly or implicitly assumed as known. This trust model analyzes the trust network to discovery trustful, influential or interesting nodes by incorporating the notion of influence together with the freshness of the trust connections among users.

Finally, there exist some approaches that have been tested on real dataset, as in our proposal. For instance, SoRec [40] is based on a factor analysis approach based on probabilistic matrix factorization to solve the data sparsity and poor prediction accuracy problems based on users' social network information and rating records. The proposal in [41] represents an approach aimed at capturing local and global social relations taking benefits of both local and global social context for recommendation. To capture the local social context, it assumes that the user preferences of two connected users are correlated by social correlation theories and, therefore, ratings from users with high reputations are more likely to be trustworthy. To capture the global social context, it uses user's reputation scores to weight the relevance of their ratings.

3. TECHNICAL PRELIMINARIES

Our scenario is represented by a virtual community $\mathcal{V}$, formally denoted as $\mathcal{V} = \langle \mathcal{A}, \mathcal{G}, \mathbf{m}, \mathbf{t} \rangle$, where $\mathcal{A}$ is the set of *agents* joined with $\mathcal{V}$ and $\mathcal{G}$ is the set of *groups* contained in $\mathcal{V}$, while $\mathbf{m}$ and $\mathbf{t}$ are two mappings denoting the *matching* metric and the *trust* metric, respectively.

We also assume that each group g is managed by an administrator agent a_g. In order to characterize the interests and the preferences of each agent $a \in \mathcal{A}$ and each group $g \in G$, we assume that a *profile* p_a (resp. p_g) is associated with a (resp. g). In particular, profile p_a is defined as a list of n *property values*, i.e. $p_a = \{\rho_a^1, \rho_a^2, .., \rho_a^n\}$, where each property ρ_a^i, $i = 1, 2, .., n$ represents the value assumed by a specific aspect that characterizes the agent a in the community.

For instance, if the virtual community is a social network, a given property could represent the set of the topics in which a is interested, or the type of groups preferred by a (e.g, a can prefer to join with public groups rather than private groups). As another example, if the virtual community is a grid/cloud community, a property could represent a quantity of a resource required or offered by a, e.g. CPU, RAM, etc.

3.1. Profile Matching

The *profile matching* metric (that we will call shortly *matching* hereafter) $\mathbf{m}(a, b)$ is a mapping that receives as input two agents a and b and yields as output a real

value, ranging in the interval $[0, 1]$ and representing how much the values of the profile properties of an agent a match with the values of the corresponding properties of another agent b, where $\mathbf{m}(a, b) = 0$ (resp. 1) means that there is no matching (resp. full matching) between the profile properties of the two agents. The matching metric is *symmetric*, i.e. $\mathbf{m}(a, b) = \mathbf{m}(b, a)$, and it is computed as follows: $\mathbf{m}(a, b) = \frac{\sum_{i=1}^{n}(\rho_a^i - \rho_b^i)}{n}$, where we assume that an appropriate operator "$-$" is defined for each property ρ^i, that returns a real value in the interval $[0, 1]$, where 0 (resp. 1) means that the property value ρ_a^i in the profile of the agent a absolutely no matches (resp. fully matches) with the corresponding value ρ_b^i, and $\rho_a^i - \rho_b^i = \rho_b^i - \rho_a^i$.

Extending the above definition, the matching between an agent a and a group g is defined by averaging the matching values $\mathbf{m}(a, b)$ of a with respect to each agent b members of g: $\mathbf{m}(a, g) = \frac{\sum_{b \in g} \mathbf{m}(a,b)}{|g|}$, where $|g|$ denotes the cardinality of group g.

3.2. Trust

The *trust* metric (shortly *trust*) $\mathbf{t}(a, b)$ is a mapping that receives as input two agents a and b and yields as output a boolean value (either 0 or 1) representing the degree of trust between two agents a and b: $\mathbf{t}(a, b) = 0$ (resp. $\mathbf{t}(a, b) = 1$) means that a assigns the minimum (resp. maximum) trustworthiness to b. The trust metric is *asymmetric*, in the sense that if a trusts b at a certain level we do not automatically expect that b trusts a at the same level and, in general, $\mathbf{t}(a, b) \neq \mathbf{t}(b, a)$.

Similarly, the trust perceived by an agent a with respect to a group g is defined as the mean of the trust values $\mathbf{t}(a, b)$ for all couples of agents $a, b \in g$, where they had some interactions in the past (therefore $\mathbf{t}(a, b) \neq NULL$):

$$\mathbf{t}(a, g) = \frac{\sum_{\{b \in g : \mathbf{t}(a,b) \neq NULL\}} \mathbf{t}(a, b)}{|\{b \in g : \mathbf{t}(a, b) \neq NULL\}|}$$

In compliance with traditional *trust theory*, we assume that $\mathbf{t}(a, b)$ is composed by two components, the first representing the *reliability* that a assigns to b in consequence of the direct experience made in past interactions, the second representing the *reputation* that b has in the community.

3.2.1. Reliability

As for the reliability, we denote it by the mapping $\mathbf{rel}(a, b)$, assuming values ranging in the domain $[0..1] \cup \{NULL\}$, while $\mathbf{rel}(a, b) = NULL$ means that a did not have past interactions with b and thus it is not able to evaluate b's trustworthiness, and the higher $\mathbf{rel}(a, b)$, the higher the perception of the reliability of b by a.

3.2.2. Reputation

As for the reputation of b, we denote it by rep_b in the interval $[0, 1] \in \mathbb{R}$.

In order to compute the reputation, we adopt the notion of *local reputation* defined in [7]. Following that approach, let $G = \langle N, A \rangle$ be a directed unlabeled graph associated with the virtual community $\mathcal{V}$, where N is a set of nodes and A is a set of arcs. Each node $n \in N$ is associated with an agent $a_n \in \mathcal{A}$ of $\mathcal{V}$, while each arc $a \in A$ is a pair (x, y), with $x, y \in N$ representing a reliability link existing in $\mathcal{V}$ between the agents a_x and a_y (i.e. $\mathbf{t}(a_x, a_y) \neq NULL$). Moreover, let $n(a)$ be the node of the graph corresponding to the agent a.

The *ego-network* of an agent a will be defined as the sub-graph of G, denoted by $G_a = \langle T, P \rangle$, where T is a set of nodes containing $n(a)$ and of all the nodes $n(k) \in A$ connected to $n(a)$, while P includes all the arcs belonging to the paths existing between $n(a)$ and $n(k)$, for each $k \in T$. In words, G_a represents all the agents both directly and indirectly *trusted* by a. We say that a indirectly trusts an agent b if there exists an agent k which either (i) directly trusts b and a directly trusts k or (ii) trusts b and a, in turn, indirectly trusts k. Hereafter, we say that an agent b belongs to the ego-network of a if the node $n(b)$ belongs to G_a.

In the context above, we assume as *local trust* a relation LT defined on $A \times A$, such that an ordered pair of agents $t = [a, b]$ belongs to LT only when the node $n(b)$ belongs to the ego-network G_a of a. Moreover, for all the nodes $n(a)$, $n(b)$ such that $[a, b] \in LT$ we also define a (normalized) local reputation measure $\lambda(a, b)$ which represents how much the agents belonging to the ego-network G_a of a trusts b.

We compute local reputation by suitably summing the contributions (in terms of trust in b) coming by all the users k (with $k \neq a$) belonging to the ego-network of a which results to be also connected with b. Let $s(a, b)$ be this sum, and we call *local network* $L(a, b)$ the set of contributors, i.e., $L(a, b) = \{z : z \in G_a \wedge \exists (z, b) \in G_a\}$. In other words, in our proposal, each node $k \in L(a, b)$ concurs to form the local reputation $s(a, b)$ by means of a contribution represented by a real value ranging between 0 and 1, where 0 (resp. 1) indicates that k has the minimum (resp. maximum) trust in b.

More specifically, if $k \in L(a, b)$ is a user in which a directly trusts, then there exists an arc $(n(a), n(k)) \in G_a$; in this case, our model assumes that the contribution of k to $s(a, b)$ will be equal to 1. Instead, if k is indirectly trusted by a, then there exists at least one path in G_a which connects a and k. We assume that the shortest path between $n(a)$ and $n(k)$ belongs to G_u and suppose it has a length $l_{a,k}$. In this case, the contribution provided by k to the trust computation we propose will be equal to $1/2^{l_{a,k}-1}$. In computing the local reputation of v, this choice corresponds to consider as exponentially less important the contributions coming from users more distant from a in the ego-network of a. To normalize $s(a, b)$ we divide it by the maximum value of the analogous

sums $s(a,z)$, for all the $z \in A$. More formally, the formula adopted for computing the (normalized) local reputation $\mathbf{rep}(a,b)$ that a perceives about b is:

$$\mathbf{rep}(a,b) = \frac{\sum_{k \in L(a,b), k \neq a,b} \frac{1}{2^{l_{a,k}-1}}}{\max_{z \in A, z \neq a,b} \left(\sum_{h \in L(a,z), h \neq a,z} \frac{1}{2^{l_{a,h}-1}} \right)} \quad (1)$$

The two trust components reliability and reputation are integrated in a unique value in the interval $[0,1]$ to compute the mapping *trust* $\mathbf{t}(a,b)$ of a about b as follows:

$$\mathbf{t}(a,b) = \alpha_u \cdot \mathbf{rel}(a,b) + (1-\alpha_u) \cdot \mathbf{rep}(a,v) \quad (2)$$

where α_u is a real coefficient belonging to [0..1] which is set by a to weight the relevance he/she assigns to the reliability with respect to the reputation.

4. COHESION AND COMPACTNESS

In this Section, we want to introduce a measure to define how much a configuration $S = \{g_1, g_2, .., g_h\}$ of groups of a virtual community $\mathcal{V} = \langle \mathcal{A}, S, \mathbf{m}, \mathbf{t} \rangle$ can be considered as *cohesive*, i.e. how much the members of the groups are satisfied to stay in those groups.

In our vision it is reasonable to calculate the *cohesion* measure on the whole configuration S, and not limited to a particular group, in other words it seems reasonable to define it by averaging an analogous measure of cohesion relative to each single group g.

Then we consider the *Average Matching*, representing the average of all the matching $\mathbf{m(a,b)}$ for each pair $a, b \in g$, and thus it appears as a reasonable manner of measuring how much the agents of g are mutually satisfied, in average, to stay in g. Therefore, a measure of the internal mutual satisfaction of the whole configuration of groups S can be computed as the *Mean Average Matching* MAM on all the groups. Formally, AM_g and MAM are defined as follows:

$$AM_g = \frac{\sum_{a,b \in g, a \neq b} \mathbf{m(a,b)}}{|g|} \quad MAM = \frac{\sum_{g \in S} AM_g}{h} \quad (3)$$

The goal is to lead any community of rational agents, starting by an initial configuration S_0 of groups at the time τ_0, to evolve in "better" configurations, improving the value of MAM until the maximum possible value. However, it is not possible to represent this goal as an optimization problem, since the property values change in time, and the best we could do is to compute the optimum configuration at a given time t. It is easy to see that find this optimum is a NP-problem. Moreover, it is not guarantee that this optimum at time t will be always the optimum of MAM also at $t+1$.

In this perspective, let S_0 be a configuration at a time t_0, then the higher the optimum of MAM at the time $t_0+\Delta$, the better the *cohesion* of the configuration S_Δ at time $t_0+\Delta$. To formally represent this notion of cohesion, we define a measure called Δ-*Cohesion* $\Psi_\Delta(S)$, defined as follows: let S be a configuration of groups in a virtual community, considered at the time t_0 and let Δ be the time-window $[t_0, t+\Delta]$. We define Δ-*Cohesion* $\Psi_\Delta(S)$ as the MAM obtained at the end of the time-window $[t_0, t+\Delta]$.

Therefore, a group configuration $S2$ having a $\Psi_\Delta(S2)$ greater than the $\Psi_\Delta(S1)$ of another group configuration $S1$, can be considered more cohesive than $S1$ in the given time-window Δ, since it finally produces groups that in average present better group matching values. In the next subsection we introduce a measure to drive a group formation strategy that leads to configurations with high group matching values.

4.1. Compactness: A measure for introducing a strategy for increasing groups' cohesion

Basing on the goal defined in the previous section, we would define a rational strategy for leading the agents of the community to change in time the configuration of groups in order to maximize the Δ-cohesion. Intuitively, in order to change the composition of the groups, agents that make their choices, i.e. to change their own group, should consider not only the matching value with each candidate group, but also the trust it has in that group. Analogously, groups should evaluate whether an agent a, which is applying to join with it, should be accepted or not, based on the trust that the group itself, as a whole, has in that agent.

For instance, let a be an agent belonging to a group g_1, and let us assume that a evaluates the possibility to change group by joining with g_2, since it evaluates that $\mathbf{m}(a,g_2) > \mathbf{m}(a,g_1)$. However, it is possible that, due to the move of a from group g_1 to group g_2 the cohesion of g_2 will change to lower values than before. Therefore, the decision of a of changing group is evaluated as not good. In this case, a could be led to make bad choices by some unreliable of even fraudulent agents, that at a given time t exhibit property values that goodly match with those of a, but that will change in the future in a undesired way.

In order to follow the above observation, in [5] we have proposed to use a suitable measure to take into account both matching and trust to evaluate the *convenience* for a to be in the same group with b. We denoted this convenience as $\mathbf{c}(a,b) = \omega \cdot \mathbf{m}(a,b) + (1-\omega) \cdot \mathbf{t}(a,b)$, where ω is a real number, ranging in $[0,1]$. In other words, we have defined the convenience as a weighted mean between matching and trust, leaving to the parameter ω the role of weighting the importance given to the matching with respect to the trust. Based on such a measure, we can introduce the *Average Convenience* AC_g of a group g, by averaging all the

convenience values for all the members of g, and the *Mean Average Convenience* MAC as follows:

$$AC_g = \frac{\sum_{a,b \in g, a \neq b} \mathbf{c(a,b)}}{|g|} \qquad MAC = \frac{\sum_{g \in S} AC_g}{h} \quad (4)$$

Analogously to the MAM, also the value of the MAC depends on the initial configuration of the groups, and on the time-window Δ in which it is computed. Therefore, we define the measure called Δ-*Compactness* $\Phi_\Delta(S)$, defined as follows: let S be a configuration of groups in a virtual community, considered at the time t_0 and let Δ be the time-window $[t_0, t_0 + \Delta]$. We define Δ-*Compactness* $\Phi_\Delta(S)$ as the MAC obtained at the end of the time-window.

4.2. Forming Cohesive Groups

The practical problem we will face in the experiments presented in Section 6 can be described by using the metrics defined in the previous subsections, as follows. If a virtual community starts by an initial configuration of groups S_0 at the time t_0, what is the configuration $S_{\delta 1}$ that we could considered as more cohesive at time $t_{\delta 1}$, i.e. at the end of the time-window $\Delta 1 = [t_0, t_{\delta 1}]$? The right answer is: "the configuration S_1 corresponding to the cohesion $\Psi_{\Delta 1}(S_0)$". However, if our temporal perspective is not time $t_{\delta 1}$, but a next time $t_{\delta 2} > t_{\delta 1}$, such that at time $t_{\delta 1}$ we want to find the configuration that we would consider as the most cohesive at time $t_{\delta 2}$, what is now the right answer? We could naturally respond "the configuration S_2 corresponding to the cohesion $\Psi_{\Delta 2}$", where $\Delta 2 = [t_0, t_{\delta 2}]$, but the problem is that if we pose this question at time $t_{\delta 1}$, we are not able to compute $\Psi_{\Delta 2}$, since we ignore how the community will evolve in the time window $\Delta 2 - \Delta 1 = [t_{\delta 1}, t_{\delta 2}]$.

Roughly speaking, in practical situations, we can construct our groups having available a *training phase* $\Delta 1$, but at the end of this phase we would desire to have a group configuration that will result the best cohesive at the end of a *test phase* $\Delta 2 - \Delta 1$. In this case, it is not guaranteed that the configuration corresponding to the cohesion $\Psi_{\Delta 1}$ will correspond to the most cohesive configuration at time $\Delta 2$. This is due to the fact that we have uncertainty about the evolution of the agents' behaviors in the unknown time-window $\Delta 2 - \Delta 1$, and we could be deceiving when forming our groups in the training phase by the behavior of unreliable agents. To avoid this problem, a reasonable solution could be to form the groups in the training phase using the compactness rather then the cohesion, since the compactness will take into account information about the agents' trustworthiness. Therefore, the configuration S_1^*, corresponding to the compactness $\Phi_{\Delta 1}(S_0)$ after the training phase, could produce a better cohesion $\Psi_{\Delta 2 - \Delta 1}(S_1^*)$ at the end of the test phase than the cohesion $\Psi_{\Delta 2 - \Delta 1}(S_1)$, produced by the configuration $S1$.

Data: u: a user, X: a set of groups, m: an integer in $[0, n]$, $k_{\texttt{MAX}}$: the number of groups u can join with
Result: A set Z of groups
Let Y be a set of m random groups extracted from DF;
Let $Z = X \bigcup Y$;
for $g \in Y$ **do**
  a_u sends a message to a_g associated with the group g and let p_g be the profile associated with g ;
end
Let $\mathcal{S}$ be the set of $k_{\texttt{MAX}}$ groups of Z having the highest values of MAC;
for $g \in \mathcal{S}$ **do**
  if $g \notin X$ **then**
    a_u sends a join request to the agent a_g that also contains the profile p_u of u;
    else
      a_u deletes u from g;
    end
  end
end
return Z

Algorithm 1: The U2G algorithm – User Agent Task

5. MATCHING USERS WITH GROUPS

In this section we sketch the design of the algorithm User-To-Group (*U2G*) presented in [5], which enables user agents to select the groups to join with by maximizing the values of compactness introduced in subsection 4.1.

Let $\mathcal{G} = \{g_1, g_2, \ldots, g_n\}$ be the groups in the virtual community, with $|\mathcal{G}| = n$. Moreover, let $k_{\texttt{MAX}}$ be a threshold ranging in $[0, n]$ which specifies the upper bound on the number of groups each user u desires to join with (for sake of simplicity, we assume that this threshold is equal for all the users of the social community).

Algorithm U2G has been designed to select $k_{\texttt{MAX}}$ groups yielding the largest value of the MAC computed on $\mathcal{G}$. We assume that as u joins with more than one group then each of them still continues to give the whole benefit to u, so that the overall benefit, in term of MAC, received by u is equal to the sum of each contribution. Therefore, in presence of an arbitrary number of groups $\mathcal{K} \subseteq \mathcal{G}$, the benefit gained by u in joining with all the groups in $\mathcal{K}$ is given by $\sum_{g_i \in \mathcal{K}} \gamma_{u \to g_i}$. The question of finding the subset $\mathcal{K}^\star \subseteq \mathcal{G}$ producing the best benefit for u under the constraint $|\mathcal{K}^\star| = k^u_{\texttt{MAX}}$ is equivalent to solve an optimization problem. While we analyzed the theoretical foundation of the problem above in work [5], in this work it is enough to say that: (i) a_u is able to sample m random

groups from $\mathcal{G}$; (*ii*) a_u will record into an internal cache the profiles of the groups u joined in the past; we shall denote this set as X; (*iii*) m is the number of the group agents that at each epoch must be contacted by a_u.

Algorithm 1 describes the steps a_u performs to find the $k_{\texttt{MAX}}$ groups to which u can join, while the corresponding algorithm implemented by the group agent can be found in [5]. In particular, it is assumed that the size of each group $g \in \mathcal{G}$ can not be bigger than a threshold $n_{\texttt{MAX}}$, $n_{\texttt{MAX}}$ is fixed by the group administrator and each agent a_g stores into an internal cache the profiles of the users who joined g.

6. EXPERIMENTS

In this section we discuss some experimental results obtained by the execution of the algorithm U2G on real datasets.

The used datasets, extracted from the social networks CIAO and EPINIONS, have been described in [41] and can be downloaded at [8]. CIAO (resp. EPINIONS) dataset consists in a matrix with a total of about 36k (resp. 922k) rows, each of them representing an event in the virtual community, in the form {*userID, productID, categoryID, rating*, helpfulness, *timestamp*}. More in detail, CategoryID represents the commercial category of the product for which the user has released a rating, and the helpfulness (a number between 1 and 6) represents the level of satisfaction of the other users for that rating. In addition, a dataset representing trust relationships is available. It consists in a list of pairs of user IDs (u, v), where each of them represents a trust relationship among user u and user v.

As discussed in Section 4.1, compactness is calculated as linear combination between trust and matching while trust, in turns, is computed as a combination of reliability and reputation, basing on Eq. 2. In our experiments, we have associated with each user a profile containing, as unique feature, the *expertise* of the user in reviewing products. This expertise has been computed by averaging, on all the reviews posted by the user, the *helpfulness* associated with each review, where the helpfulness is an information available on the dataset and obtained by the opinions expressed by the users of the community.

Instead, the reliability is represented by the values found in the dataset of trust relationships, while reputation has been calculated based on Eq. 1 (Section 3.2.2). In particular, reputation has been computed by considering trust relationships until the second level.

Experiments were performed as follows:

- rows of the dataset are *arranged in increasing order*, basing on the timestamp found in the sixth column (rating timestamp);
- the dataset is divided into 11 time-windows Δ (see Section 4), such that the first time-window is used as training set, the remaining ten are used for the subsequent tests; the first time-window comprises about the first 30% of events while the remaining events (about $35k$) are divided among the remaining time-windows;
- the reliability matrix is constructed by loading the dataset containing trust relationships once and for all;
- the *training* is performed by executing the algorithm *U2G* on the first time-window Δ_1. At the end of this phase, a cohesion Ψ_{Δ_1} is measured;
- the test phase is performed by computing subsequent cohesion values, by adding data of time-windows $\Delta_2, \ldots, \Delta_{end}$, until the final value $\Psi_{\Delta_{end}}$ is found.

Overall, comparing the starting value of cohesion Ψ_{Δ_1} – calculated after the training phase, which drives group formation – and the final one, $\Psi_{\Delta_{end}}$ – calculated after the test phase – is useful to understand the ability, over time, to form cohesive groups, of a certain algorithm (e.g. U2G), based on a certain criteria. Furthermore, as our goal is to understand the contribution or, in other words, the ability to form cohesive groups on the basis of the trust measure, particularly its components, reliability and reputation, we are particularly interested in comparing final values of cohesion, i.e. $\Psi_{\Delta_{end}}$, among the different criterias used to perform the training test (e.g. matching only vs reliability).

To this end, we performed a number of experiments that can be divided into three categories, based on the criterias used to perform the training phase: training based on matching (m), training based on matching and trust (mr to mrp), and training based on trust only (r to rp), as shown in table 1. Columns ω and α represent the parameters used to configure the different experiments: ω is the weight assigned to the matching in the computation of the Compactness, therefore $\omega = 1$ means that only matching is considered, while $\omega = 0$ means that only the trust contribution is actually weighted in the computation of compactness; α is the weight which balance trust and reputation, based on equation 2.

Parameter LR (fifth column in table 1) is a flag for discriminating the case in which we use the local reputation from the case we use only the reliability. Thus, in order to investigate in the contribution of local reputation, we used a variation of equation 2, as follows. If $LR == 1$ (see Table 1), we consider a trust using both local reputation and reliability, weighted by α. In the case reliability is zero, we use only the local reputation. This variation is necessary because in the case of CIAO and EPINIONS, reliability is a boolean value, and reliability equal to 0 for an agent does not discriminate the case in which we do not trust the agent from the case we do not know the agent. Therefore, in this case, the following computation was performed:

TABLE 1. Summary of the experimental results

No.	**T** (Training)	**Meaning**	ω	α	**LR**	CIAO		EPINIONS	
						Ψ_{Δ_1}	$\Psi_{\Delta_{end}}$	Ψ_{Δ_1}	$\Psi_{\Delta_{end}}$
1	m	only matching	1	–	–	0.69	0.74	0.85	0.83
2	mr	matching and reliability	0.5	1	0	0.68	0.74	0.85	0.83
3	mrr	matching and reliability+reputation	0.5	0.5	1	0.69	0.73	0.84	0.83
4	mrp	matching and reputation	0.5	0	1	0.70	0.74	0.86	0.84
5	r	reliability	0	1	0	0.71	0.78	0.85	0.84
6	rr	reliability+reputation	0	0.5	1	0.60	0.80	0.86	0.85
7	rp	reputation	0	0	1	0.69	0.78	0.85	0.85

(LR=Local Reputation Flag activated)

$$\mathbf{t}(a,b) = \begin{cases} \alpha\ rel(a,b) + (1-\alpha)\ rep(a,v) & \mathbf{rel}(1,b) \neq 0 \\ rep(a,bv) & \mathbf{rel}(1,b) = 0 \end{cases} \quad (5)$$

in case $LR == 0$, the contribution of the local reputation is not considered, therefore

$$\mathbf{t}(a,b) = \alpha\ rel(a,b) \quad (6)$$

Finally, we remember that values of reliability are taken from the values of trust found in the dataset, which can be zero or one. Hence, when $LR == 0$, $t(a,b)$ will assume values 0 or α. Columns no. 6-7 and 8-9 of table 1 shows values Ψ_{Δ_1}-$\Psi_{\Delta_{end}}$ obtained for CIAO and EPINIONS respectively.

6.1. Evaluation

In the following we compare values of $\Psi_{\Delta_{end}}$ of cases $T = \{mr, mrr, mrp\}$ (rows No. 2-4 of table 1) with that obtained for $T = m$ (row No. 1 of table 1). This case is labeled *matching only vs [matching + trust]*. Finally, we compare values of $\Psi_{\Delta_{end}}$ obtained for $T = \{r, rr, rp\}$ (rows No. 5-7 of table 1) with that obtained for the case m. This case is labeled *Matching only vs Trust*. The reader may refer also to figure 1, on which we report all the values of $\Psi_{\Delta_{end}}$ collected for CIAO and EPINIONS.

Matching only vs [Matching + Trust]. In this first set of experiments we have compared the final cohesion of groups when the training is performed by considering only the matching criteria ($T = m$), and that obtained by mixing matching and trust ($T = \{mr, mrr, mrp\}$). First of all, it can be observed that the training based on the combination of matching and reliability ($T = mr$) will form groups with $\Psi_{\Delta_{end}}$ very similar to those formed by means of the only matching ($T = m$). Therefore, the first result is represented by the fact that forming groups by considering also the reliability does not degrade the cohesion of the groups (cfr. figure 1). A further experiment is represented by the case $T = mrr$, on which the training is still performed on the base of matching and trust, and the trust component is represented by a mix of reliability and local reputation. Also in this case, it can be observed that the contribution given by the reputation does not lead negative changes of the final cohesion at the end of the test phase, i.e. $\Psi_{\Delta_{end}}$. One step forward in this investigation is represented by $T = mrp$, a further experiment on which groups are formed by means of a training based on the mix between matching and local reputation. In this case we observe that using the local reputation does not lead negative changes of the final cohesion (cfr. figure 1). In other words, local reputation can be used in place of reliability when groups are formed by mixing matching and trust.

Matching only vs Trust. Now we compare the value of $\Psi_{\Delta_{end}}$ obtained for matching only ($T = m$), with that obtained for $T = \{r, rr, rp\}$. In this case, by setting parameter $\omega = 0$, only trust is included in the computation of compactness, used in the training phase, in order to from groups. The first observation is that final values of cohesion $\Psi_{\Delta_{end}}$ are, in overall, larger than values obtained in the previous cases $T = \{mr, mrr, mrp\}$ for CIAO, and almost identical than in the previous case for EPINIONS (cfr. figure 1). In particular, we started the experiments by using only the reliability value (r), by setting $LR = 0$ and $\alpha = 1$. In this case we don't observe degradation in the cohesion of groups. Instead, even a little improvement of about 5% is obtained for CIAO, if compared with the value for $T = m$ (cfr. figure 1). By the subsequent experiment – $T = rp$ (reputation only)– we can conclude that local reputation, i.e. suggestions given by friends and friends of friends, can be effective in forming cohesive groups as much as direct knowledge ($T = r$), as it gives almost identical value of cohesion (cfr. figure 1). A further interesting result is represented by the case $T = rr$, which shows the best value of $\Psi_{\Delta_{end}}$, with a gain of about 8% in the case of CIAO, and also in the case of EPINIONS we have a little improvement of about 3%, as shown in figure 1.

7. CONCLUSION

Most of the work on group formation in virtual communities propose to use some measure of profile matching to aggregate agents having similar requirements. However, comparing profiles is not always possible in social

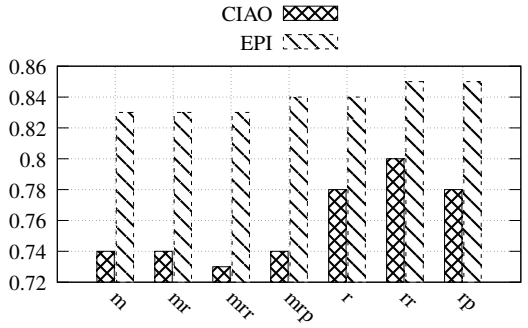

FIGURE 1. Values of $\Psi_{\Delta_{end}}$ for $T = \{m, \ldots, rp\}$

environments, where agents are human users that often prefer to make private their personal information. Moreover, comparing user profiles is a computationally hard task in a large virtual community, that could be composed by millions of agents. More recent work introduced the idea of combining trust measures with the traditional profile matching measures, to compensate possible insufficient profile information. However, the question left open is that of understanding if the introduction of the trust is concretely advantageous on real virtual community. In this work, we have defined a theoretical framework to face such a problem, and we have applied it to two well-known online social networks, often used in the past as benchmarks, namely CIAO and EPINIONS. More precisely, we propose to represent the attitude of a group to maintain its internal homogeneity in a time interval Δ by a measure called Δ-cohesion, only based on profile matching. Then, we have defined another measure, mixing profile matching and trust, that we have denoted as compactness. Finally, we have tried to form groups on the real social networks of reference by optimizing, at time t_0 the compactness, comparing our results with those obtained forming groups by optimizing only the profile matching. In both cases, the results are represented in terms of Δ-cohesion. Moreover we have considered two different types of trust measures, namely the reliability, i.e. the direct trust that an agent has in another one, based on past direct interactions, and the local reputation, i.e. the trust that an agent perceives about another one based on some recommendations coming from her/his neighborhood of friends. From the experiments presented in Section 6, we can conclude that, in order to form cohesive groups, the computation of the profile matching matrix is even not needed, as the sole reliability allows groups administrators and agents to organize groups which have the same cohesion than those formed on the base of matching criteria. Furthermore, another important obtained result is the following: if information about reliability is not provided, local reputation will give similar performances. The best result is obtained by mixing in equal measure local reputation and reliability. Our ongoing research is now devoted on giving a theoretical interpretation of these results. The intuition lead us to argue that trust, and in particular local reputation, is a powerful tool to substitute profile matching for forming cohesive groups, but it is now necessary to characterize in which measure the property of the trust network influence the effectiveness of such a substitution. This is the main goal of our future work.

ACKNOWLEDGEMENTS

This work has been partially supported by the following projects: (*i*) CLARA funded by the Italian Ministry of Education, University, and Research; (*ii*) *Programma Operativo Nazionale Ricerca e Competitivitá* 2007-2013, project BA2Kno (Business Analytics toKnow),in *Laboratorio in Rete di ServiceInnovation* and (*iii*) *Programma Operativo Nazionale Ricerca e Competitivitá* 2007-2013, Distretto Tecnologico CyberSecurity funded by the Italian Ministry of Education, University and Research.

REFERENCES

[1] Facebook (2014). https://www.facebook.com/.

[2] Twitter (2014). https://www.twitter/.

[3] Foster, I., Kesselman, C., and Tuecke, S. (2001) The anatomy of the grid: Enabling scalable virtual organizations. *International journal of high performance computing applications*, **15**, 200–222.

[4] Grozev, N. and Buyya, R. (2014) Inter-cloud architectures and application brokering: taxonomy and survey. *Software: Practice and Experience*, **44**, 369–390.

[5] Meo, P. D., Ferrara, E., Rosaci, D., and Sarnè, G. (2015) Trust and compactness in social network groups. *ACM Transactions on Cybernetics*, **45**, 205–2016.

[6] Bonhard, P., Harries, C., McCarthy, J., and Sasse, M. A. (2006) Accounting for taste: Using profile similarity to improve recommender systems. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, April CHI '06, pp. 1057–1066. ACM.

[7] De Meo, P., Messina, F., Rosaci, D., and Sarné, G. M. L. (2014) Recommending Users in Social Networks by Integrating Local and Global Reputation. In Fortino, G., Di Fatta, G., Li, W., Ochoa, S., Cuzzocrea, A., and Pathan, M. (eds.), *Internet and Distributed Computing Systems: 7th International Conference, IDCS 2014, Calabria, Italy, September 22-24, 2014. Proceedings.* Springer International Publishing, Cham.

[8] Tang, J. (2017). Trust/distrust computing.

[9] Rathnayaka, A. D., Potdar, V. M., Dillon, T. S., and Kuruppu, S. (2014) Formation of virtual community groups to manage prosumers in smart grids. *International Journal of Grid and Utility Computing*, **6**, 47–56.

[10] Pandey, M., Pathak, V. K., and Chaudhary, B. D. (2012) A framework for interest-based community evolution and sharing of latent knowledge. *International Journal of Grid and Utility Computing*, **3**, 200–213.

[11] Huang, Y., Bessis, N., Kuonen, P., and Hirsbrunner, B. (2011) Casp: a community-aware scheduling protocol. *International Journal of Grid and Utility Computing*, **2**, 11–24.

[12] French, T., Bessis, N., Xhafa, F., and Maple, C. (2011) Towards a corporate governance trust agent scoring model for collaborative virtual organisations. *International Journal of Grid and Utility Computing*, **2**, 98–108.

[13] Andrews, S. and Orphanides, C. (2012) Knowledge discovery through creating formal contexts. *International Journal of Space-Based and Situated Computing*, **2**, 123–138.

[14] Hsu, T.-Y. and Kshemkalyani, A. D. (2015) Variable social vector clocks for exploring user interactions in social communication networks. *International Journal of Space-Based and Situated Computing*, **5**, 39–52.

[15] Heidemann, J., Klier, M., and Probst, F. (2012) Online social networks: A survey of a global phenomenon. *Computer Networks*, **56**, 3866–3878.

[16] Zhan, J. and Fang, X. (2011) Social computing: the state of the art. *International Journal of Social Computing and Cyber-Physical Systems*, **1**, 1–12.

[17] Fogel, J. and Nehmad, E. (2009) Internet social network communities: Risk taking, trust, and privacy concerns. *Computers in Human Behavior*, **25**, 153–160.

[18] Aikebaier, A., Enokido, T., and Takizawa, M. (2011) Trustworthy group making algorithm in distributed systems. *Human-centric Computing and Information Sciences*, **1**, 6.

[19] Gurjar, K. and Moon, Y.-S. (2016) Comparative study of evaluating the trustworthiness of data based on data provenance. *Journal of Information Processing Systems*, **12**, 234–248.

[20] Elshaafi, H. and Botvich, D. (2013) Trustworthiness inference of multi-tenant component services in service compositions. *JoC*, **4**, 31–37.

[21] Amer-Yahia, S., Roy, S., Chawlat, A., Das, G., and Yu, C. (2009) Group recommendation: Semantics and efficiency. *Proc. of the VLDB Endowment*, **2**, 754–765.

[22] Baltrunas, L., Makcinskas, T., and Ricci, F. (2010) Group recommendations with rank aggregation and collaborative filtering. *Proceedings of the Fourth ACM Conference on Recommender Systems*, New York, NY, USA, September RecSys '10, pp. 119–126. ACM.

[23] Liu, X., Datta, A., Rzadca, K., and Lim, E.-P. (2009) Stereotrust: a group based personalized trust model. *Proc. of the 18th ACM conf. on Information and knowledge management*, New York, NY, USA, November, pp. 7–16. ACM.

[24] Chen, W., Zhang, D., and Chang, E. (2008) Combinational collaborative filtering for personalized community rrecommendation. *Proc. of the ACM Int. Conf. on Knowledge Discovery and Data mining (SIGKDD'08)*, pp. 115–123. ACM, New York, NY, USA.

[25] Vasuki, V., Natarajan, N., Lu, Z., Savas, B., and Dhillon, I. (2011) Scalable affiliation recommendation using auxiliary networks. *ACM Transactions on Intelligent Systems and Technology*, **3**, 3:1–3:20.

[26] A.A.V.V. (2013). Advogato's trust metric. `http://www.advogato.org/trust-metric.html`.

[27] Golbeck, J. and Hendler, J. (2006) Inferring binary trust relationships in web-based social networks. *ACM Transactions on Internet Technology*, **6**, 497–529.

[28] Guha, R., Kumar, R., Raghavan, P., and Tomkins, A. (2004) Propagation of trust and distrust. *Proc. of the 13th International Conference on World Wide Web*, New York, NY, USA, May, pp. 403–412. ACM.

[29] Huynh, T., Jennings, N., and Shadbolt, N. (2006) An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, **13**, 119–154.

[30] Dellarocas, C. (2010) Designing reputation systems for the social web. *SSRN Electronic Journal*, **18**. Available at https:ssrn.comabstract=1624697.

[31] Kim, Y. and Song, H. (2011) Strategies for predicting local trust based on trust propagation in social networks. *Knowledge-Based Systems*, **24**, 1360–1371.

[32] Rosaci, D., Sarnè, G., and Garruzzo, S. (2012) Integrating trust measures in multiagent systems. *International Journal of Intelligent Systems*, **27**, 1–15.

[33] Pinyol, I. and Sabater-Mir, J. (2013) Computational trust and reputation models for open multi-agent systems: a review. *Artificial Intelligence Review*, **40**, 1–25.

[34] Massa, P. and Avesani, P. (2007) Trust metrics on controversial users: Balancing between tyranny of the majority. *International Journal on Semantic Web and Information Systems (IJSWIS)*, **3**, 39–64.

[35] Ziegler, C. and Lausen, G. (2004) Spreading activation models for trust propagation. *e-Technology, e-Commerce and e-Service, 2004. EEE'04. 2004 IEEE International Conference on*, USA, March, pp. 83–97. IEEE.

[36] Golbeck, J. (2005) Computing and applying trust in web-based social networks. PhD thesis University of Maryland, Department of Computer Science College Park.

[37] Massa, P. and Avesani, P. (2007) Trust-aware recommender systems. *Proc. of the 2007 ACM Conference on Recommender systems*, New York, NY, USA, October, pp. 17–24. ACM.

[38] Jamali, M. and Ester, M. (2009) Trustwalker: a random walk model for combining trust-based and item-based recommendation. *Proc. of the 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, New York, NY, USA, june, pp. 397–406. ACM.

[39] Varlamis, I., Eirinaki, M., and Louta, M. (2013) Application of Social Network Metrics to a Trust-Aware Collaborative Model for Generating Personalized User Recommendations. In Özyer, T., Rokne, J., Wagner, G., and Reuser, A. H. (eds.), *The Influence of Technology on Social Network Analysis and Mining*. Springer Vienna, Vienna.

[40] Ma, H., Yang, H., Lyu, M. R., and King, I. (2008) Sorec: Social recommendation using probabilistic matrix factorization. *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, New York, NY, USA, october CIKM '08, pp. 931–940. ACM.

[41] Tang, J., Hu, X., Gao, H., and Liu, H. (2013) Exploiting local and global social context for recommendation. *Proceedings of the Twenty-Third International Joint*

Conference on Artificial Intelligence, Palo Alto, California, August IJCAI '13, pp. 2712–2718. AAAI Press.