Scalable Computing: Practice and Experience
Volume 9, Number 1, pp. 29–37. http://www.scpe.org
ISSN 1895-1767

THE DILEMMA OF TRUST: A SOCIAL NETWORK BASED APPROACH

V. CARCHIOLO, A. LONGHEU, M. MALGERI, G. MANGIONI, V. NICOSIA*

Abstract. Trust is essential in human interaction and this naturally reflects within computer science context, in particular inside internet-based communities. In this work we present an approach based on social networks, which revealed several useful properties, as the *small world* effect, that can be usefully exploited in addressing the question of trust. We aim at reproducing the behaviour individuals adopt in their life when they establish and maintain trust relationships, sending queries to collect reputations in order to estimate how much trust in new acquaintances. We also consider issues as query forwarding and lifecycle of trusting relationships, aiming at building an effective and efficient model for trust management.

Key words. trusting and reputation, social networks, p2p networks

1. Introduction. Trust is a crucial matter in several human interactions that must be addressed in order to cope with uncertainty when living and working with others. The question naturally reflects within the computer science context, specifically inside Internet-based communities, where the spread of networks and sharing of information over last decades (as in P2P paradigma) imposes to address the dilemma of trust, that can be viewed as an extension of trusting among people.

Several different definitions can be found for trust, e.g. in [13, 17, 1, 5]; a simple and effective definition is "trust provides information about with whom we should share information, from whom we should accept information, and what consideration such information should be given".

Studies on social networks over last years revealed that relationships are characterized by useful properties, as the "small world" effect, according to which a generic person is connected to another one through short paths of social relationships. This mechanism can be positively exploited when addressing the question of trust, in particular by emulating the behaviour individuals adopt in the real world to evaluate how much to trust others.

In our work, we consider how a person joins an existing network of trust relationships in real life, and how he builds these relationships over time, in particular reproducing the mechanism of collecting and mediating other's opinion about an unacquainted person to assign him a trust value. Our proposal falls within the context of *reputation-based* systems [7, 3], allowing an estimation of trust, in opposition to *policy-based* systems, where the hard evidence of owned credentials is used to grant trust[1]. We also consider trust relationships lifecycle, i.e. the evolution from "weak" (first time impressions) to "strong" (experience based judgement) links, as well as the removal of links between individuals not having contacts for long periods, and the effect of "feedback", that is the propagation of trust value from a person to neighbours, in order to inform them about a positive or negative achieved judgement about a node.

In summary, our approach is to exploit the reputation about a node A, which is actually distributed among A's neighbours in the network in order to achieve an average value for this reputation to be suggested to a new node when he has to assign a trust level to A. Our reputation-based system is used to evaluate trust, hence to modify trusting network over time, also exploiting feedback mechanisms.

The paper is organized as follows: in section 2 we describe how a person interacts with an existing trust-based (i.e. friendship) community with whom he is initially unacquainted, formalizing our discussions into a trusting algorithm; trust links lifecycle are introduced in section 3, whereas in sections 4 preliminary results of network simulation are shown. Finally, in sections 5 and 6 we respectively present related works and our conclusions.

2. The mechanism of trusting within social networks.

2.1. Concepts and Terms. According to the definition given in [13], we say a person (Alice) trusts another one (Bob) if A is persuaded that information coming from B are true, for example, if A needs B to help her in finding a job, and B says that he will help A, A is guaranteed that B says the truth, hence he will actually help A. This definition expresses the same concepts introduced in [4], where trust is intended as the choice A makes when she perceives an ambiguous path and B's actions determine how the result of following the path will be; a similar definition is given in [17], where "trust is a bet about the future contingent actions of others". We quantify how A trusts B with a number $t_B^A \in [-1, 1]$, where -1 means A does not trust B at

* Dipartimento di Ingegneria Informatica e delle Telecomunicazioni, Facoltà di Ingegneria, Università degli Studi di Catania – Viale Andrea Doria 6, I95125, Catania, ITALY – {car,alongheu,mm,gmangioni,vnicosia}@diit.unict.it

all, and 1 indicates that A trusts B completely; a value of 0 models the *indifference*, due to either a lack of information about B or when exactly opposite judgements come from the network, hence A is unable to decide whether trust or not in B.

We will refer to trust value also as a "risk factor", i.e. the risk A accepts when she believes that B's actions will lead to a good outcome. After introducing a definition for trust, we have to consider that trust is strictly related to acquiring information, i.e. in a social network a trust relationship between A and B is established if A collects some information about B, either with a direct interaction with B or even getting such information from other sources, e.g. his acquaintances, web sites, e-mails, contact board, and so on; indeed, if A does not know anything about the existence of B, A also cannot assign any trust value to B. We name these information about B as B's "resume", representing information as study degree, working experiences, personal skills and attitudes, and so on; at this stage, we do not impose any constraint on resume's content or structure

We denote the resume about a node X owned by a node Y as:

$$C_X^Y = \{\mathsf{X}, \overline{C_X^Y}\} \tag{2.1}$$

where X is the identifier for the node X the resume is about, and $\overline{C_X^Y}$ is the set of information cited previously. Note that different nodes actually can know different things about X, as in real world occur, thus we indicate the owner node Y; moreover, resumes evolve over time, as soon as Y acquires information about X, thus updating $\overline{C_X^Y}$. The most updated and complete resume is the one owned by the person it concerns, simply indicated as $\overline{C_X}$.

2.2. A sample scenario. To model trusting mechanisms inside a community, we use a directed labeled graph $\mathcal{G}$, defined as a pair $(\mathcal{N}, \mathcal{E})$ where $\mathcal{N}$ is a set of nodes and $\mathcal{E}$ is a set of edges (or links) between vertices, i.e. $\mathcal{E} = \{(u, v) | u, v \in \mathcal{N} \wedge u \neq v\}$; a link from A to B is labelled with the pair trust, resume, i.e. (t_B^A, C_B^A)

To show how links are established, let us consider Alice (A), an unacquainted person who wants to establish some contact with an existing community, usually to satisfy some needs in her everyday life. For instance A may be a new student that needs to know where classrooms are located, hence she asks for some help to other students she meets in a corridor, or she can search for a computer science professor on the University website. In the first case, A directly contacts other individuals, say Bob (B) and Carl (C), thus having links form A to B and from A to C, whereas in the last case A collects information without a direct interaction with individuals, but she is still collecting resumes about some professors (say Danny and Eve, respectively node D and E), so links from A to D and E are present too. Note that arcs are oriented since mutual trust values are not necessarily the same, or one of them could also be not present, for instance A could completely trust in B and C, since she is in a new, unknown context, but B and C might assign a low trust value to A, being an acquaintance they are meeting for the first time; moreover, A will assign trust values to professor D and E based on their resumes, but D and E actually do not know anything about the existence of A, hence neither arc from D to A nor from E to A will be present.

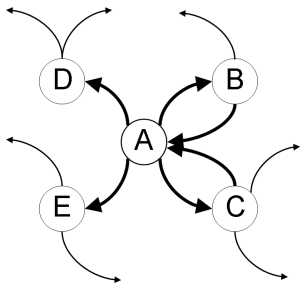


FIG. 2.1. *Node A joining the network*

In this scenario, represented in figure 2.1, where thinner lines represent trusting relationships with other nodes, A assigns a first trust value to nodes B,C,D and E, that will be refined over time, as soon as other information (resumes) is available

In this scenario, A assigns a first trust value to nodes B,C,D and E. Such values will be refined over time, as soon as other information (resumes) is available, e.g. A can re-evaluate t_D^A and t_E^A as soon as she meets those professors; similarly occurs for trust values assigned to A by nodes B and C.

2.3. Trust evaluation. In addition to direct (personal) interaction between two individuals, in the real world the refinement process of a trust value assigned by a person A to a generic person X, not belonging to the set of A's acquaintances, is often performed by simply asking to X's acquaintances, i.e. to those people who directly know X and can provide a judgement about him. These judgements, together with all resumes about X, are collected and mediated by A, so she can refine her knowledge and assign a proper trust value about X exploiting other people's personal experience with X. Note that in the graph model, the term "acquaintance" correspond to a node that is neighbour to another, i.e. a path of length 1 (an arc) exists between these nodes.

In this paper we focus on the trust refinement mechanism exploiting acquaintances opinions, since it better represents social network interactions. A, before asking information about X, setup the net by adding X, if not present, and creating an arc pointing to it labelled by $risk_A$, that represents a default value typical of A.

---

addToNet

Require: $\mathcal{G}$ is the graph representing the net
Require: $\mathsf{A}, \mathsf{X} \in \mathcal{N}$
Require: $risk_A$ is A's ingenuity
1: create a new arc in $\mathcal{N}$ from A to X: $arc(\mathsf{A}, \mathsf{X})$
2: label $arc(\mathsf{A}, \mathsf{X})$ using $\{risk_A, C_A^X\}$

The *trustNode* algorithm implements the evaluation of mediated trust performed by a generic node A about a node X; in the following we also comment relevant instructions.

---

trustNode

Require: $\mathcal{G}$
Require: $\mathsf{A}, \mathsf{X} \in \mathcal{N}$ where A is a node that wishes to trust node X
Require: C_X
Require: τ_{good} that is threshold used to select node for query forwarding
Require: n_r maximum number of request, id_q is a query identifier
1: **if** $\mathsf{X} \notin \mathbf{N}^A$ **then**
2: $addToNet(\mathsf{A},\mathsf{X})$
3: **end if**
4: $\overline{\mathbf{R}} = \{r_i \in \mathbf{R}_X^A | r_i > \tau_{good}\}$
5: $\mathbf{T}_X^A = \emptyset$
6: **for all** $(\mathsf{I} \in \overline{\mathbf{R}}) \wedge (\mid \mathbf{T}_X^A \mid < n_r)$ **do**
7: $t_I^X = forward(I, C_X, id_q)$
8: $\mathbf{T}_X^A = \mathbf{T}_X^A \bigcup \{\mathsf{I}, t_X^I\}$
9: **end for**
10: **if** $\mid \mathbf{T}_X^A \mid < n_r$ **then**
11: **for all** $(\mathsf{I} \in (\mathbf{N}^A - \overline{\mathbf{R}})) \wedge (\mid \mathbf{T}_X^A \mid < n_r)$ **do**
12: $t_X^I = forward(\mathsf{I}, C_X, \mathsf{A}, id_q)$
13: $\mathbf{T}_X^A = \mathbf{T}_X^A \bigcup \{\mathsf{I}, t_X^I\}$
14: **end for**
15: **end if**
16: $t_X^A = trust(\mathbf{T}_X^A, \mathbf{N}^A)$

First (line 1-3), A check whether X does not already belong to the set of her acquaintances (denoted as $\mathbf{N}^A$); in this case the *addToNet* function creates the arc from A to X.

Now, A would collect opinions about X to X's acquaintances, but probably she does not know anyone of them, hence A asks *her* personal acquaintances whether they know X, or whether acquaintances of A's acquaintances know X and so on, actually forwarding her request (*query*) through the network by exploiting trusting relationships. Specifically, A initially establish how many opinions (say, n_r) about X she needs to assign X a trust value: the more opinions will ask, the more important for A the evaluation is, the more accurate the trust value to assign will be; the drawback will be the flooding of messages traversing the network. To model real world interactions, A establishes an order according to acquaintances are contacted. In particular, A first

considers acquaintances she trusts more, at the same time having a resume similar to X's resume, to show why we consider resumes similarity, suppose for instance that from X's resume, A knows that X studies nuclear physics at the university, and A also suppose that trusts B and C the same, but B also is a nuclear physics student. To assign a trust value to X, in the real world A will first ask B, since A supposes that similarity in resumes increases the possibility that those acquaintances directly know X.

To formalize this discussion, we introduce the *correspondence* function to evaluate resumes similarity and *relevance* function to sort acquaintances based on correspondence and assigned trust level:

- *correspondence* is defined as $corr : C \times C \rightarrow [0; 1]$, also noted with $corr(C_I, C_X)$; we call $c_{I,X}$ the result of correspondence between nodes I and X. At this stage, the implementation of this function simply adopt classical information retrieval techniques aiming at extracting inverse index vector for both resumes. Obviously, $corr(C_I, C_X) = corr(\overline{C_I}, \overline{C_X})$.
- *relevance* function is defined as $r_I^{A,X} = c_{X,I} \cdot t_I^A$ where A is the node that ask an opinion about X to one of her neighbour I; the set of all values is:

$$\mathbf{R}_X^A = \{r_I^{A,X} | \mathsf{I} \in (\mathbf{N}^A - \{\mathsf{X}\})\} \tag{2.2}$$

Given these definitions, in line 4 of the algorithm A establishes a threshold τ_{good} to find acquaintances she both trusts a lot and also with a resume highly similar to X's. These *qualified* acquaintances are considered first (lines 6-9) when A sends the query (line 7), using the $forward$ function to collect their opinions about X, finally storing this trust value together with the neighbour that provided it in the following set (line 8):

$$\mathbf{T}_\mathsf{X}^A = \{(\mathsf{I}, t_X^A) | \mathsf{I} \in \overline{\mathbf{N}}_A\} \tag{2.3}$$

$\overline{\mathbf{N}}_A$ indicates the set of acquaintances (neighbours) who actually answered the query. Note that acquaintances with high relevance are randomly selected when sending queries in order to avoid always contacting the same acquaintances.

A starts sending queries to qualified acquaintances, aiming at getting n_r answers from them, but when their number is not enough or when the number of successfully queries is not enough (line 10), A will continue sending queries to less qualified acquaintances (lines 10-15); note that A might not get all n_r answers, as in real world occurs.

Finally, the collected answers are used to build a mediated opinion about X. When evaluating this average trust level opinions coming from acquaintances should be weighted according to trust level A assigned to them, reflecting the real world behaviour. The formula we adopt is the following:

$$t_X^A = trust(\mathbf{T}_X^A, \overline{\mathbf{N}}_A) = \frac{\sum t_X^i \cdot t_i^A}{\sum t_i^A} \tag{2.4}$$

This is essentially a weighted average formula where main terms are t_X^i, i.e. acquaintances opinions about X, weights are t_i^A, that is trust values assigned by A to his acquaintances. To understand why t_i^A is also present at the denominator, we impose that acquaintances A assigned low trust values should have less influence over the resulting mean value, in order to reflect the real world behaviour adopted by a person, i.e. he tends to neglect the opinion of an acquaintances he trusts in with a low degree. To do this, at the denominator we place the sum of *contributions*, being each acquaintance's contribution the trust value A assigned him; if A trusts with a low degree someone (i.e. related trust value tends to 0), his contribution as well as also the corresponding term at the numerator will be neglected. The t_X^A is finally associated to the arc from A to X.

A relevant question is that when each of A's acquaintances receives the query, his task is to answer the query with his opinion about X, but the current acquaintance could still get into the same trouble, i.e. he does not know directly X; similarly to A, such node will forward the request to his acquaintances (but A), waiting for a set of trust values about X to be mediated as described previously, hence the resulting mean will be assigned by the node to X and it will be also delivered to A (or generally to the requesting neighbour node). The propagation of the query will occur until one of X's acquaintances is reached. The $forward$ function uses an algorithm (not reported here) very similar to the $trustNode$, except for two mechanisms:

1. Similarly to the first node that generated the query (A), an *intermediate* node needs n_r opinions about X, but differently from A, if n_r is not achieved exploiting his qualified acquaintances, no more answers will be asked to acquaintances having a relevance below the threshold, modeling the absence of a personal interest the intermediate node has in determining an opinion about X. In the worst case, the intermediate node will give back a trust value of 0 (i.e. he will answer he cannot give a significant opinion about X). The n_r and τ_{good} are the same A established; this models the propagation of the *importance* A assigned to the query.
2. The formula adopted by each intermediate node to determine the mediated trust value is the same adopted by A, but the trust value the intermediate node returns is multiplied by a distance factor, i.e. a value in the range]0,1] that models the fact that in the real world we consider more significant an opinion if it comes from a person that directly knows X, decreasing the value of this opinion as soon as increases the number of individuals to be contacted to get to this opinion.

3. Trusting relationships lifecycle. Social networks dinamically evolves by changing both their arcs (i.e. trust values) and nodes (coming and leaving the trusting network), so topology between requests changes. In the following we describe such situations.

3.1. Feedback. Acting as in social networks, whenever a node B behaves positively (good actions), any node A who assigned a trust level to B should raise her reputation about B, lowering trust in the case of B perpetrates some bad action, hence any trust value about someone should be always subject to changes whenever we get information about that person. Note that if this does not occur, this would denote that A is not capable of processing external information, as it happens for some diseases involving brain damages; however, we do not focus on these situations, hence we suppose that any information about individuals coming form the external world is processed and leads to re-evaluations of associated trust levels.

Re-evaluation of trust about B could also be triggered by A itself if the trust level was evaluated a long time ago; we name *aging* this situation (see next paragraph for more details).

No matter which is the reason, whenever a node A evaluates or re-evaluates a reputation about a node B and consequently adjust B's trust level, according to the algorithm illustrated in section 2, A should also spread this new information over the network, in order to inform about B's behaviour (possibly different from the past); we call *feedback* this mechanism. Usually A cannot inform the entire network, rather he/she propagates the information to his neighbours (but B), allowing them to adjust their opinion about B. The set of neighbours considered is $\overline{\mathbf{N}}_A$, i.e. the set of acquaintances (neighbours) who answered the re-evaluation query. Among these neighbours, A considers the subset of those satisfying the following conditions:

1. their trust value is over a given threshold, i.e. A strongly trusts them,
2. the trust level about B they gave to A in the past is very different from the reputation evaluated by A; note that A got the vector of neighbours' judgements about B during the re-evalution process. itself

Both conditions aim at modelling the behaviour within social networks, in particular A considers strongly trusted neighbours *worth* to be adviced about her trust level about B (first condition), and inform them only if this is useful (second condition). Also note that A could extends this feedback also to other neighbours, e.g. those belonging to $\overline{\mathbf{N}}_A$ sets used in past (not the last) B reputation evaluation process; this depends on how much *memory* about past evaluation A retains; in our approach, we consider just the last one.

Another issue to address concerns the behaviour of one of A's acquaintances (say C) as soon as she receives the new trust value about B, for instance if C receives a strong negative feedback from A about B and C assigned a high trust level to B and a lower to A, should C decrease his trust on B (i.e. should he believe to A's judgement?) Or should he decrease his judgement about A' (since she provided an unaffordable opinion)? Or should C decrease both?

Our choice is always inspired by real world behaviour, hence as soon as node C receives B's trust level, he first consider his trust level about A; if this is below a given threshold, he will not consider information coming from A to modify his opinion about B neither perform any propagation. If C strongly trusts A, he:

1. uses A's trust value about B to re-evaluate his trust level about B; note that this does not trigger the algorithm introduced in section 2, rather C simply introduces A's judgement about B into the formula 2.4, substituting previous A's contribution if A was already included in $\overline{\mathbf{N}}_A$, or properly extending $\overline{\mathbf{N}}_A$ otherwise.
2. once a new updated judgment about B has been evaluated, C can stop or he can further propagate this information to a subset of his neighbours, according to the same criterion adopted by A previously.

We choose to propagate, so the information can be actually spread over the trust network; note that in order to avoid excessive network traffic, we adopt a simple solution similar to the *distance factor* introduced in the previous section, i.e. we consider the inverse of hops number (from A) involved in the propagation process, and a node along this chain will stop only when this factor will be lower than a given threshold.

3. C does not change his opinion about A since in this case he trusts her enough to consider her judgement about B as valid; in other words, it would be inconsistent a modification over C's opinion about A.

Feedback mechanism we introduced heavily influences trusts, infact modifing trust values too often could make the net unstable whereas infrequent modifications would not permit an accurate perception of "vox populi" about nodes. In summary, the feedback mechanism reflects the real world behaviour that occurs when A evaluates a more recent opinion about a person and wishes to communicate this opinion to all of her trusted acquaintances, since she thinks that this opinion is more affordable than other existing opinions her acquaintances currently own. Further works should be devoted to a deep analysis of threshold influences the feedback mechanism.

3.2. Weak and Strong links. Each time a node A collects information about an unacquaintance node X and consequently assigns X an initial trust value without involving his acquaintances, we name the arc from A to X as *weak* link, since the value of trust has not been verified neither by A's personal experience, nor by collecting acquaintances opinions; as soon as A has a personal experience with X or she gets other opinions and build a refined trust value, the arc will be named as *strong*. A weak link is used to model the first contact a person has with one another, a contact that implies the assignment of a first, temporary and possibly wrong trust level, as in real world actually occurs; for instance, all links A creates with nodes B,C,D and E in section 2 are initially weak.

A strong link can be lowered to a weak one when its associated trust level has been evaluated a very long time ago, hence that link should be not considered affordable anymore, and a new evaluation should be performed. Similarly, if a weak link does not become strong for a long period, it will be removed from the network, modeling the fact that A forgets about X existence if he never contact X during his life. This *aging* mechanism is also useful to avoid a network with an excessive (and probably useless) links.

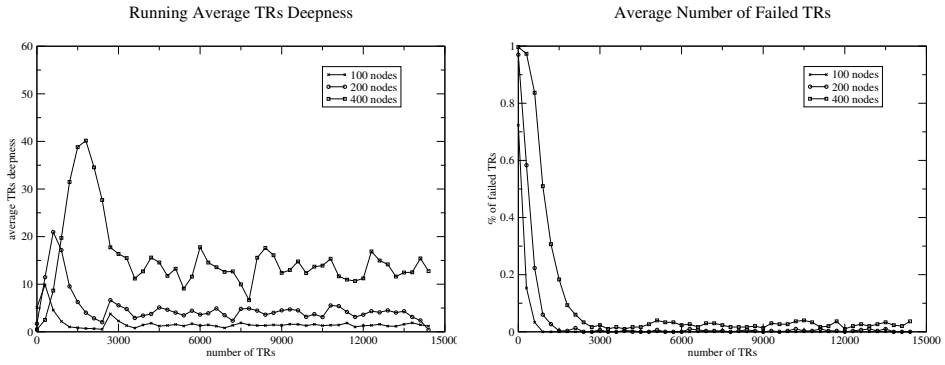
Finally, note that the mechanism of evaluating trusting by collecting opinions from acquaintances relies both on weak and strong links.

4. Preliminary results. A first set of simulation has been performed using the trusting algorithm described previously, in addition with aging mechanism, in order to test the convergence of the network. Simulation have been performed considering networks with different number of nodes (i.e. 100, 200 and 400), with initially no links; the trusting algorithm is invoked when queries are generated, and it allows the growing of links number, hence the network evolves through a transient state until a stable set of connections is achieved. Input parameters we assigned for trusting algorithm are $n_r = 3$ (in order to avoid the flooding of the network), and distance factor=0.9 (to allow long paths for query forwarding to be considered). Results are quantified by introducing two measures:

1. average trusting requests deepness, denoted as $ATRD$ (where a TR is a query), indicating the deepest path from the requesting node to the node to be trusted, mediated among all TRs; this property is used to express the *efficiency* of the trusting process
2. average number of failed TRs ($AFTR$), where a TR is considered *failed* when no answers are collected, e.g. when no relevant nodes are found for TR forwarding. This property is used to express the *effectiveness* of the trusting process

In fig. 4.1(a) and 4.1(b) running $ATRD$ and $AFTR$ are represented, respectively. These express how many TR are needed for the network to converge, i.e. to provide stable values for both properties. This two experiments clearly point out that the proposed trusting algorithm converge and, in particular, TRs deepness reachs stable values when about 1500–3000 (for networks of 100 to 400 nodes, respectively) TRs are performed. Note that TRs deepness initially increases since when the network is created too few links are present; as soon as links number increases, the deepness decreases since more (possibly shorter) paths are available. The number of failed TRs rapidly decrease to zero when the network converges, meaning that when a trusting request is issued, exactly n_r opinions are received from the source node allowing it trust evaluation according to eq. 2.4.

In addition to $ATRD$ and $AFTR$ in this paper we shortly discuss the algorithm resilience to group empowering. This measure shows which is the critical cardinality of a cheating group to be able to raise (or equivalently



(a) Average TRs deepness (b) Average number of failed TRs

FIG. 4.1. *Network evolution*

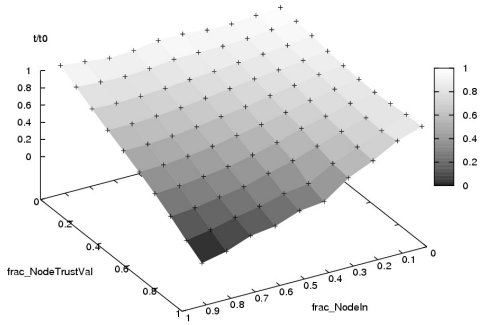


FIG. 4.2. *Resilience to group empowering*

lower) the reputation of a target peer. The experiment is conduct on a network of 100 nodes where 15000 TRs are performed (as previously discussed, in this case the network reaches a stable state). Then, we randomly choose a source node S, a target node T and a TR from S to T is executed; the corresponding trust level $t_0{}_T^S$ is used as a comparison factor during the subsequent TRs. This experiment is repeated lowering from 0% to 100% the trust value ($frac\_NodeTrustVal$ in figure 4.2) of a given fraction ($frac\_NodeIn$ in figure 4.2) of the number of nodes having T as neighbour (n ranges from 1 to $|\mathbf{P}^{\mathsf{T}}|$, $\mathbf{P}^{\mathsf{T}} = \{\mathsf{U} \in \mathcal{N} | (\mathsf{U}, \mathsf{T}) \in \mathcal{E}\}$).

For a given pair ($frac\_NodeTrustVal$, $frac\_NodeIn$), we evaluate t_T^S hence the ratio $t_T^S / t_0{}_T^S$. The experiment is iterated changing source and target nodes in order to obtain the average real network behaviour. The ratio gives an idea about the algorithm robustness against group empowering; indeed, as shown in figure 4.2, a group of cheating nodes can heavly affect the judgement on a target node (i.e. lowering $t_T^S / t_0{}_T^S$) only when $frac\_NodeIn$ is a significant fraction (about 40-50%) over $|\mathbf{P}^{\mathsf{T}}|$, or when $frac\_NodeTrustVal$ grows up to about 40%.

5. Related work. Trust has been studied in social sciences, business and psychology before it became central to computer science research; the survey [1] offers a complete overview on most recent issues concerning trust.

In [10], the first work providing a formal model of trust, the value of trust is chosen within the range [-1,1], covering from complete distrust to full trust. We follow the same approach, since we believe this is the best representation: simple, normalized and symmetric around the zero (indifference). As in [10] is claimed, probably none of the extremes (i.e. full trust or distrust) is actually possible. In [16] however, the use of a range with negative values to model distrust is somehow criticized, mainly due to algorithmic-related issues (no more necessarily real values for trust matrix eigenvector); authors also debate about whether a trust score of 0 translate to distrust or to *no opinion*; we use the [-1,1] range since negative values provide the right adjustment when evaluating reputation of a given node.

The paper [11] shows some similarities with our work. First, authors also view the question of trust in the computer science context as a propagation of social matter ("in today's connected world, it is possible and common to interact with unknown people ..."). The metric they define for trust evaluation is based on their previous work, MoleTrust, that predicts the trust score of source user on target user by walking the social network starting from the source user and by propagating trust along trust edges; intuitively the trust score of a user depends on the trust statements of other users on her and their trust scores. This approach is quite similar to our proposal, except for some aspects:

- to stop walking on the network, authors define the *trust propagation horizon*, which specifies the maximum distance from source user to which trust is propagated along trust chains. We instead claim that our *distance factor* is a better choice since it gracefully decreases the importance of nodes along the walk, whereas with the horizon the walk is interrupted.
- for predicting the trust score of a user, MoleTrust analyzes the incoming trust edges and discards the ones coming from users with a predicted trust score less than 0.6 *threshold*; our approach allows a finer granularity in deciding trusted nodes by exploiting the *relevance*.
- both approaches share the difficulty of applying the local trust metrics to real available data set as Epinions.com, since they often accept only binary trust values instead of continuous (but more realistic) values.
- Finally, in the MoleTrust-based approach, it is deeply analyzed the role of controversial users, i.e. those who are judged in very different ways by other nodes. In particular, they show that a local trust metric (i.e. trust assigned to a node depends on the point of view of the evaluating node) is better than a global one since it significantly reduces the prediction error for controversial users. Our approach put both positive and negative opinions in the same formula when evaluating reputation, hence it is not required to distinguish opinions.

In the last decades, the recommender system is gaining an important role in today's networked worlds because they provide tool to support decision helping in selecting reliable from unreliable. Reliability is often expressed through a trust value with which each agent labels its neighbours; [14, 9] explore this, but they does not investigate in the topic of formation of trust based on real-world social studies. Some recent works have suggested to combine distributed recommendation systems with trust and reputation mechanisms [14, 12].

People surfing the Web has already faced the matter to form opinion and rate them against trust, as far as the well-known reviewers' community *Epinions (*`http://www.epinions.com`*)* which collects reviews from the community members, any members can also decide to "trust" or "distrust" each other. All the trust and block relationships interact and form a hierarchy known as the Web of Trust. This Web of Trust (WOT) is combined with rating to determine in what order opinions are shown to readers. However trusting model remains centralized (trust is influenced only by the manager).

Ziegler and Golbeck [6, 18] believe that computational trust model bear several favorable properties from social filtering than they expect notions of trust must reflect user similarity. Therefore a reputation system is an important tool in every network, but assume a central role in emerging P2P networks where many people interacts with many others in a sort of "democratic" fashion. Some author discusses decentralized methods that approximate ranks according to local history and a notion of neighbourhood [2] where trust is calculated trying advantage of small-world properties often emerging in networks that mimic real world. In P2P area EigenTrust [8] propose a distributed implementation of PageRank [15] that needs also a distributed structure to store data and imposes to pre-trust all nodes belonging to the net thus reducing the "de-centralisation". The forward algorithm introduced in previous section proposes a mechanism for trusting propagation; a little taxonomy of other possibile propagation criteria can be found in [16]. Our approach is someway similar to their *direct propagation* mechanism; we believe others mechanisms described seem to be a limited validity.

In [18], they expect notions of trust to clearly reflect user similarity; this is similar to our *relevance* and *correspondence* factors, whose aim is indeed to take into account the similarity of resumes.

6. Conclusions and future work. This paper introduced an approach to trust entities hint at human behaviour; we proposed an algorithm based on local behaviour of an human and we believe that several important properties of social network will emerge. The algorithm has been discussed and motivated thruoghout the paper; preliminary promising results about the convergence of the network and its resilience to group empowering have been shown.

Further studies are currently active on testing network behaviour, as well as on investigation about communities that may emerge—of interest, similarity, confidence, etc—and on the importance of *correlation* among resumes to improve the effectiveness and efficiency of the proposed approach.

REFERENCES

[1] D. Artz and Y. Gil, *A survey of trust in computer science and the semantic web*, Web Semantics: Science, Services and Agents on the World Wide Web, 5 (2007), pp. 58–71.

[2] M. Dell'Amico, *Neighbourhood maps: Decentralised ranking in small-world p2p networks*, in 3rd International Workshop on Hot Topics in Peer-to-Peer Systems (Hot-P2P), Rhodes Island, Greece, April 2006.

[3] Z. Despotovic and K. Aberer, *P2P reputation management: probabilistic estimation vs. social networks*, Comput. Networks, 50 (2006), pp. 485–500.

[4] M. Deutsch, *Cooperation and trust, some theoretical notes*, in Nebraska Symposium on MOtivation, N. U. Press, ed., 1962.

[5] J. Golbeck, *Trust and nuanced profile similarity in online social networks.* ACM Transactions on the Web, to appear.

[6] J. Golbeck and J. Hendler, *Reputation network analysis for email filtering*, in Conference on Email and Anti-Spam (CEAS), Mountain View, CA, USA, July 2004.

[7] M. Gupta, P. Judge, and M. Ammar, *A reputation system for peer-to-peer networks*, (2003), pp. 144–152.

[8] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, *The eigentrust algorithm for reputation management in P2P networks*, in In Proceedings of the Twelfth International World Wide Web Conference, 2003.

[9] M. Kinateder and S. Pearson, *A Privacy-Enhanced Peer-to-Peer Reputation System*, in Proceedings of the 4th International Conference on Electronic Commerce and Web Technologies (EC-Web 2003), K. Bauknecht, A. M. Tjoa, and G. Quirchmayr, eds., vol. 2738 of LNCS, Prague, Czech Republic, September 2003, Springer-Verlag, pp. 206–215.

[10] S. Marsh, *Formalising trust as a computational concept*, tech. report, Department of Mathematics and Computer Science, University of Stirling, 1994.

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

[12] P. Massa, B. Bhattacharjee, *Using trust in recommender systems: An experimental analysis*, in iTrust, 2004, pp. 221–235.

[13] F. C. Mish, ed., *Dictionary and Thesaurus*, Merriam-Webster, Incorporated, 2007.

[14] M. Montaner, B. Lopez and J. L. de la Rosa, *Opinion-based filtering through trust*, in CIA '02: Proceedings of the 6th International Workshop on Cooperative Information Agents VI, London, UK, 2002, Springer-Verlag, pp. 164–178.

[15] L. Page, S. Brin, R. Motwani, and T. Winograd, *The pagerank citation ranking: Bringing order to the web*, tech. report, Stanford Digital Library Technologies Project, 1998.

[16] P. Raghavan, R. Guha, R. Kumar, and A. Tomkins, *Propagation of trust and distrust*, in Proc. of WWW2004 conf., 2004.

[17] P. Sztompka, *Trust: A sociological theory*, (1999).

[18] C.-N. Ziegler and J. Golbeck, *Investigating correlations of trust and interest similarity—do birds of a feather really flock together?*, Decision Support System, (2005).

Edited by: Maria Ganzha, Marcin Paprzycki
Received: Jan 31, 2008
Accepted: Feb 9, 2008