# Comparing Local, Collective, and Global Trust Models

Charif Haydar, Azim Roussanaly, Anne Boyer

# Comparing Local, Collective, and Global Trust Models

Charif Haydar
Université de Lorraine
Laboratoire Loria, Nancy, France.
Email: charif.alchiekhhaydar@loria.fr

Azim Roussanaly
Université de Lorraine
Laboratoire Loria, Nancy, France.
Email: azim.roussanaly@loria.fr

Anne Boyer
Université de Lorraine
Laboratoire Loria, Nancy, France.
Email: anne.boyer@loria.fr

***Abstract*—Today, trust modelling is a serious issue on the social web. Social web allows information exchange between anonymous users who have no prior knowledge to each other. The aim of a trust model is to rerank acquired information according to their reliability and the trustworthiness of their author. During the last decade, trust models were proposed to assist the user to state his opinion about the acquired information, and about their sources. We identify three paradigms for trust modelling: the first relies on evaluating previous interactions with the source (individual trust), the second relies on the word of mouth paradigm where the user relies on the knowledge of his friends and their friends (collective trust), and the third relies on the reputation of the source (global trust). In this paper, we propose and compare three trust models, each of which represent one of the precedent paradigms. All three models make use of subjective logic (SL). SL is an extension of probabilistic logic that deals with the cases of lack of evidence. It supplies framework for modelling trust on the web. The comparison includes three axes: the precision, the complexity and the robustness to malicious attacks. We show that each of the three models has a weak point in one of the three axes.**

***Keywords*—Trust modelling, Subjective logic, Collective trust, global trust, local trust, reputation**

## I. Introduction

Web 1.0 provided a popular access to the largest data store ever existed (Internet). The major difficulty resided in extracting relevant information and resources from the huge mass of data available for most queries. Information retrieval (IR) came out to yield Internet more efficient and exploitable by ranking resources according to their relevance to queries. Then, web 2.0 arrived with more interactive tools such as forums and social networks. The numerous people who were only the spectators in web 1.0, became the actors in web 2.0. They are now able to share their own opinions and knowledge. Collaborative IR and social recommender systems (RS) [34] are now used to rank this kind of resources.

Web 2.0 provides a highly connected social environment. It allows data exchange among anonymous people from all around the world. Acquiring information from such sources raises the question about its reliability and trustworthiness. Modelling social trust into computational trust appeared to overcome the trustworthiness problem (for both information and resources). Today, computational trust is integrated in many domains and contexts such as social networks, recommender systems [4], [25], file sharing [22], multi-agents systems [31] etc.

We consider social trust as the belief of an individual, called truster, that another individual, called trustee, has the competence and the willingness to either execute a task to the favour of the truster, or to assist him to execute it. The assistance can simply be recommending another individual to execute the task. The truster tries to acquire information and constructs his own belief about the trustee before deciding to cooperate with him [1].

Building truster's opinion about the trustee is mainly derived by three means; the first is by exploiting previous interactions between both of them, so the truster relies on his own knowledge about the trustee (individual opinion). The second uses the word of mouth mechanism, where the truster exploits the collective knowledge of his trustee friends and their friends (collective opinion). The third is by relying on a global reputation score associated to the trustee (global opinions).

Our objective in this paper is to propose and compare three trust models based on the three types of opinions. A local trust model that uses the individual opinions when they are available, and collective opinions otherwise. A collective trust model that uses strictly collective opinions. A global trust model that uses only global opinions. We evaluate these three models from the perspective of precision, complexity, and robustness to malicious attacks. All our models use a framework of subjective logic (SL) [17], which is an extension of probabilistic logic, based on the belief theory [24], [23]. SL provides a flexible framework form modelling trust [1], [2].

The object of our comparison is the dataset stackexchange [16]. It is a social website based on a question answering platform to assist users to find answers to their questions in diverse domains (programming, mathematics, English language, cooking, etc.). We assume that proposing an answer is a proof of willingness to assist the person asking. Therefore, our objective is to find the user capable to provide the most relevant answer.

The paper is organized as follows: in Section II, we present the general framework, starting by presenting social trust and computational trust. In II-C, we introduce subjective logic and some of its operators. In Section III, we detail the three proposed models. In Section IV, we describe the used dataset, and present our interpretation of the success and the failure of an interaction according to current data structure. In Section V, we discuss the results of the three axes of comparison. Finally, in Section VI, we resume our conclusions and future work.

## II. General Framework

The objective of trust is to find the appropriate person to cooperate with in order to achieve a given task. Truster's decision about to cooperate or not is influenced by many factors such as: the context, the completeness of his opinion about the trustee, the reputation of the trustee, the emergency of the task for him, and many more. In the following section, we present a real life example about trust in order to explain this phenomena, and some factors that can influence the cooperation decision.

Suppose that Alice wants to paint her house. She publishes this information and receives three offers from three professional candidates (Eric, Fred and George) *willing* to do the job for her. She already knows Eric because he painted her clinic sometime ago. Alice does not know neither Fred nor George. If Alice is satisfied by the job of Eric in her clinic, she might hire him for the house directly, and ignore the offers of Fred and George. Nevertheless, if Alice is perfectionist, she will investigate about them. Alice can ask her friends (Bob and Caroline) about Fred and George. She also might use a referential organization that classifies painters, or any other mean to acquire informations about the reputation of the three painters.

Suppose that Bob says that Fred is a good professional. Caroline says that she recently hired George to paint her house and she is not satisfied about his work, whereas her sister Diana has hired Fred and was satisfied. Note that even though Alice trusts Bob and Caroline, she will not ask any of them to paint her house, because she thinks that they *lack competence* in this domain. Even so, they are still capable to play an important role as advisers or recommenders.

After the suggestions of Bob and Caroline, Alice will eliminate George and choose between Eric and Fred.

In this scenario, Alice asked her friends only about the candidates that she herself does not know. But the scenario could have been changed if she asked them also about Eric. Bob could say for example that Eric is good for concrete walls used in Alice's clinic, but he is not very competent for wooden walls like those of Alice's house. This information can be sufficient to convince Alice to hire Fred instead of Eric.

This example shows the limit of direct interactions manner, and that the word of mouth may be useful to enrich the knowledge of the truster about the trustee. It can lead to sharpen his decision even when he thinks that his own acquired knowledge is sufficient to take a decision.

In another scenario, Alice could simply search for the best ranked painter referenced by specialised magazine, syndicate, or other organization. Usually, these rankers track all the interactions of their target, and use his entire history to perform their ranking. As we can see in fig. 1, neither local nor collective trust model would allow Alice to get use of the interaction of Henry with Fred, as no path connects her to Henry. The global trust models use the opinions of all the users about Fred regardless if Alice trust them or not. Global opinions are based on a larger number of interactions. Note that the active user has no control on the users who participate in building this kind of opinions for him. His own opinion about participants is not considered.

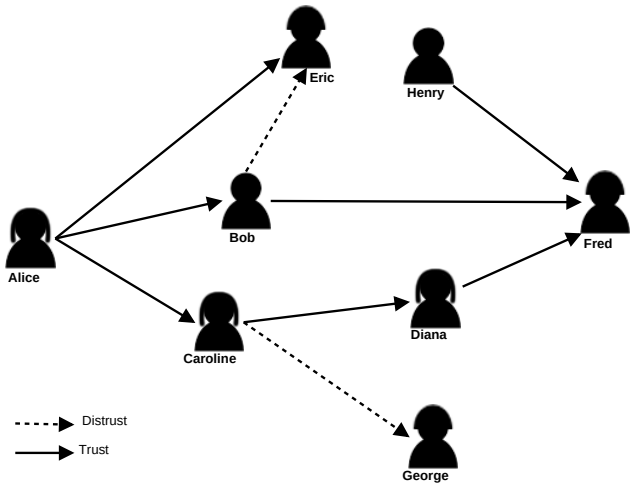


Fig. 1: Trust network

Furthermore, the current example allows us to distinguish four types of trust relationships; these types are also discussed in [28]:

1) Direct trust: trust is the result of interactions between exclusively the truster and trustee, such as the relations "Alice Bob" and "Alice Eric".
2) Indirect trust: the two persons do not know each other. Trust is established due to trustee intermediate persons, such as the relation "Alice Fred".
3) Functional trust: the expectation of the truster is that the trustee accomplishes the task himself, such as the relation "Alice Eric", "Alice Fred" and "Alice George".
4) Referential trust: the expectation of the truster is that the trustee will recommend someone to accomplish the task, such as the relation "Alice Bob" and "Alice Caroline". Note that the recommendation of Caroline is also based on her referential trust in her sister Diana. In other words, no obligation for the trustee in referential trust to base his recommendation on a functional trust relation. Normally, a series of referential trust relations must end with one functional trust relation [29].

Fig. 1 illustrates the trust network used by Alice to make her decision.

In the next section, we discuss the formalization of social trust for the social web, and compare the different models that exist.

### A. Computational trust

Computational trust raised in the last decade to ensure trust awareness in intelligent systems. It usually consists of a formalization of social trust adjusted to specific context and application. Basically, computational trust has three axes [18]:

- Quantitative, also called global-trust or reputation: the system computes a score for each user, this score represents his global trustworthiness. This score is

considered when any other user needs to interact with this user [20].

- Qualitative, also called local-trust or relationship: it takes into account the personal bias. It is represented as user to user relationship. It is the trustworthiness of a user Y from the point of view of one single user X [20].
- Process driven (system): it represents the trust of the users in the system [18].

This work focuses on the qualitative and quantitative axes. Most local trust models [4], [21], [19], [26] tend to formulate local trust problem in the form of a trust network. A trust network is a directed weighted graph where vertices represent users, and edges represent trust relationships. Models differ by their notation of edges, and their strategies in traversing the network to compute trust between two unconnected users. This operation is called Trust propagation. It is fundamental in local trust models, as it allows to estimate how much a user $A$ (called source node) should trust a user $B$ (called destination node).

Global trust models [32], [22] associate a score of reputation to each user. This same score is used in all the interactions where this user is implicated as a trustee. These models do not take the personal bias into consideration, so a reputed user is reputed to everybody and vice versa.

Local trust models suffer from a cold start problem, they can not deal neither with new users nor with users having no friends [2]. Global trust models are not concerned by this problem. Nevertheless, it is difficult for new users to build their own reputation in a global trust model, since ancient reputed users are usually more susceptible to be recommended by the system.

As most social applications, social recommender systems are exposed to different types of malicious attacks [14], [33]. Malicious attackers aim to take the control over the recommender system for different purposes, such as driving the system to recommend or to oppose to the recommendation of given items, inserting viruses, spam or advertises, etc.

Trust-aware recommenders are more robust than other recommenders for most attacks [35]. Nevertheless, they are not completely immune to all kinds of malicious attacks, such as group attacks [36] which is always possible in some trust models.

Computational trust is applied to many fields in artificial intelligence, recommender systems, file sharing, Spam detection, networks security, etc. Most computational models are fitted to their application fields and context. Basically, we identify two categories. Models dealing only with trust relationships, and models dealing with trust and distrust relationships.

The first category contains numerous models such as [7], [3], [8], [5], [6], [9]. The main disadvantage of this category is that models do not distinguish between distrusted and unknown persons. Social systems have to give chances to new and unknown users to prove their trustworthiness, whereas it must be more severe in blocking distrusted and malicious users [15]. Unknown users are often new users, a system unable to distinguish them from distrusted users risk to be very severe with them, so discourage the evolution of the trust network, or to be so tolerant even with distrusted users, so less efficient.

Models in the second category distinguish between unknown and distrusted people. Models in [11], [12], [13], [28], [10], identify three possible cases: trust, distrust and ignorance. Authors in [10] classify these models into two groups; gradual models [11], [12], [10] and probabilistic models [13], [28]. Gradual representation of trust is more similar to the human way in expressing trust, whereas probabilistic representation is more meaningful mathematically.

We use SL [28], [17] in our models. Our choice is motivated by many factors. SL considers trust ignorance and distrust relationships, which is compatible with our need to distinguish between unknown and distrusted people. Most other trust models consider the creation and the evolution of trust links as an external issue, they describe and deal with existing links. SL is more transparent about this issue, trust relationships in SL are based on the accumulation of interactions between a couple of users. It proposes many operators that allow to integrate many aspects and factors of trust, which make it one of the most generic and flexible trust models.

It is based on the belief theory [24], [23], which offers the capacity to aggregate many beliefs coming from many sources (even contradictory ones), which corresponds to the case when a user has to aggregate the opinions of many friends of him about a given problem.

Nevertheless, we compare them to referential model called MoleTrust [4]. This model has been frequently used in the trust based recommendation, and proved its quality in this domain, and surpassed the collaborative filtering in the term of performance. We explain it in the following Section II-B, before proceeding to the Section II-C which is dedicated to explain the structure and some operators of subjective logic.

### B. MoleTrust

Moletrust was presented in {massa04. It considers that each user has a domain of trust, where he adds his trustee friends to. User can either fully trust other user or not trust him at all. The model considers that trust is partially transitive, so its value decline according to the distance between the source user and the destination user. The only initializing parameter is the maximal propagation distance $d$.

If user $A$ added user $B$ to his domain, and $B$ added $C$, then the trust of $A$ in $C$ is given by the equation:

$$Tr(A,C) = \begin{cases} \frac{(d-n+1)}{d} & \text{if } n \leq d \\ 0 & \text{if } n > d \end{cases} \quad (1)$$

Where $n$ is the distance between $A$ and $C$ ($n = 2$ as there two steps between them; first step from $A$ to $B$, and the second from $B$ to $C$).
$d$ is the maximal propagation distance.

Consider $d = 4$ then: $Tr(A,C) = (4 - 2 + 1)/4 = 0.75$.

We consider that when a user $A$ accepts an answer of another user $B$, that $A$ trust $B$. A Moletrust link between both

users is created. While the algorithm is not aware to distrust so no interpretation exists for unaccepted answers.

### C. Subjective logic

Subjective logic (SL) [17] is an extension of probabilistic logic, which associates each probability with a degree of uncertainty. Subjective logic allows to build models that treat with situations of incomplete evidences.

Belief theory [24], [23] is a special case of probability theory dedicated to treat incomplete knowledge. The sum of probabilities of possible cases can be less than 1. Subjective logic [27] offers a belief calculus using a belief metrics called opinion. The opinion of an individual $U$ about a statement $x$ is denoted by:

$$\omega_x^U = (b, d, u, a)$$

where: $b, d, u \in [0, 1]$ are respectively the belief, disbelief and uncertainty of $U$ about $x$. The sum of the three values equals to one (i.e $b + d + u = 1$). Base rate $a \in [0, 1]$ is the prior probability. Basically, base rate is a statistical measure applied in cases of evidences' absence. For example, when we know that the percentage of a disease $x$ in a given population is $1\%$, then the base rate of $x$'s infection is $1\%$. When we meet a new individual who did not make a test for the disease, a priori we assume that the probability that he is infected is $1\%$. In social trust cases, while no a priori statistics are present, we consider that unknown person has a half chance to be trustworthy. So we use a base rate $a = 0.5$. In subjective logic, the base rate steers the contribution of the uncertainty in the computation of the probability expectation value according to 2:

$$E(\omega_x^U) = b + a \times u \tag{2}$$

The opinion in subjective logic is based on the accumulation of successful and failed experiences. After each experience, $U$ updates his opinion about $x$ consistently with experience's outcome. According to this description, opinion can be represented as a binary random variable. Beta distribution is normally used to model the behaviour of this kind of variables. By consequence, the opinion corresponds to the probability density function (PDF) of beta distribution. PDF is denoted by two evidence parameters $\alpha$ and $\beta$ that can be written as functions of the number of successful and failed experiences respectively.

$$\begin{aligned} \alpha &= r + W \times a \\ \beta &= s + W \times (1 - a) \end{aligned} \tag{3}$$

where $r$ is the number of successful experiences (evidences). $s$ is the number of failed experiences. $W$ is the non-informative prior weight that ensures that the prior (i.e., when $r = s = 0$) Beta PDF with default base rate $a = 0.5$ is a uniform PDF (normally $W = 2$).

The expectation value of beta PDF is:

$$E(Beta(p|\alpha, \beta)) = \frac{\alpha}{\alpha + \beta} = \frac{r + Wa}{r + s + W} \tag{4}$$

In subjective logic, the mapping between the opinion parameters and the beta PDF parameters is given as follows:

$$b = \frac{r}{(r + s + W)} \tag{5}$$

$$d = \frac{s}{(r + s + W)} \tag{6}$$

$$u = \frac{W}{(r + s + W)} \tag{7}$$

Table I shows an example of the evolution of an opinion with successive interactions.

TABLE I: Opinion evolution with successive interactions

| No | state | r | s | belief | disbelief | uncertainty |
|---|---|---|---|---|---|---|
| 0 | no interaction | 0 | 0 | 0 | 0 | 1 |
| 1 | successful interaction | 1 | 0 | 1/3 | 0 | 2/3 |
| 2 | failed interaction | 1 | 1 | 1/4 | 1/4 | 2/4 |
| 3 | successful interaction | 2 | 1 | 2/5 | 1/5 | 2/5 |

In the first line of Table I, we see the case of absence of evidences (experiences). The opinion is completely uncertain $(u = 1)$. In this case, according to 2, the expectation value equals to the base rate value. The arrival of new experiences, will make the uncertainty decreases, regardless if these experiences are successful or failed. Successful experiences will augment the belief, whereas failed experiences will augment the disbelief.

Subjective logic opinions can be illustrated in the interior of an equilateral triangle. The three vertices of the triangle are called belief, disbelief, and uncertainty. The uncertainty axis links the uncertainty vertex with the opposite edge (the belief-disbelief edge), the uncertainty value of the opinion is plotted on this axis considering that its contact with the edge belief-disbelief represents the value 0, whereas the contact with the uncertainty vertex represents the value 1. In the same way, we describe the belief and the disbelief axis.

The opinion is represented by the intersection point of the three projections on the three axis (belief, disbelief and certainty) as shown in the example in Fig. 2. The bottom of the triangle is the probability axis, the probability expectation value is the projection of the opinion point on the probability axis with respect to the line linking the uncertainty vertex with the base rate point on the probability axis. Fig. 2 illustrates an example of opinion mapping in subjective logic. The opinion is represented by a point inside the triangle. The point is the intersection of the projection of the three values $b$, $d$, and $u$ on the axis of belief disbelief and uncertainty, respectively. the probability expectation value $E(x)$ is the projection of $\omega_x$ on the probability axis directed by the axis linking $a_x$ with the uncertainty edge.

Note that changing the value of base rate can make people more reckless or more cautious.

After defining the structure of the opinion in subjective logic, we need to explain some of subjective logic operators that are useful for building trust network. Local trust networks are usually represented by a direct graph, where vertices represent users, and edges represent trust relations. Consequently, computing trust value between two users is reduced to finding a path or more connecting them to each other.

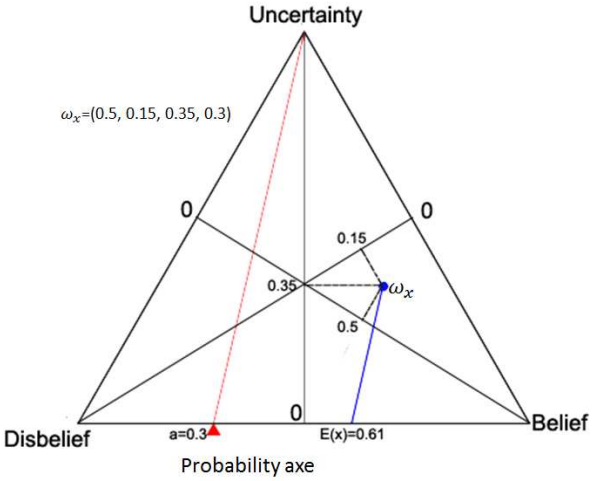


Fig. 2: Subjective logic Opinion

*1) Trust transitivity:* If an individual $A$ trusts another individual $B$, and $B$ trusts $C$, trust transitivity operator is used to derive the relation between $A$ and $C$.

Subjective logic proposes the uncertainty favouring transitivity. This operator enable the user $A$ to receive the opinion of a friend $C$ of his trustee friend $B$, or to ignore the opinion of $B$ in case of $A$ distrust $B$. Formally the operator is given by (8).

$$\omega_B^A = b_B^A, d_B^A, u_B^A, a_B^A$$

$$\omega_C^B = b_C^B, d_C^B, u_C^B, a_C^B$$

$$\omega_B^A \otimes \omega_C^B = \begin{cases} b_C^{A:B} = b_B^A.b_C^B \\ d_C^{A:B} = b_B^A.d_C^B \\ u_C^{A:B} = d_B^A + u_B^A + b_B^A.u_C^B \\ a_C^{A:B} = a_C^B \end{cases} \quad (8)$$

*2) Opinion fusion:* Suppose in the previous example that $A$ has another trustee friend $D$ who also trusts $C$. $A$ has two separate sources of information about $C$.

Subjective logic proposes two main types to fuse $B$'s and $D$'s opinions about $C$:

$$\omega_B^C \underline{\oplus} \omega_D^C = \begin{cases} b_{B\diamond D}^C = \frac{b_B^C.u_D^C + b_D^C.u_B^C}{u_B^C + u_D^C - u_B^C.u_D^C} \\ d_{B\diamond D}^C = \frac{b_B^C.u_D^C + b_D^C.u_B^C}{u_B^C + u_D^C - u_B^C.u_D^C} \\ u_{B\diamond D}^C = \frac{u_B^C + u_D^C}{u_B^C + u_D^C - u_B^C.u_D^C} \end{cases} \quad (9)$$

This operator allows the user to aggregate the opinions of his trustee friends, regardless if their opinions were contradictory or not.

## III. Proposed Models

The aim of our models is to predict the most relevant answer to a given question within a list of answers. Basically, trust models consider that the person asking tends more to accept answers written by trustworthy people, so trust models try to retrieve these users. We have developed three trust aware models. All of them are based on subjective logic. We refer to them as local trust model (LTM), which is a classical local trust model, so it exploits only individual opinions when they are available, otherwise it exploits collective opinions. Collective trust model (CTM) which exploits collective opinions all the time, and global trust model (GTM), which depends on context-aware reputation scores.

### A. Local trust model

This model is basically based on the model proposed in [28]. It consists of building a local trust network between users. The edges of this network are SL opinions of users about each other. Formally, we represent the trust network as a graph $G = (V, E)$, where $V$ represents the set of vertices (users), and $E$ represents the set of edges (direct trust relationships). Suppose that a user $a$ asks a question $q$, a set of users $\mathcal{R}$ will propose many answers to him. The aim of the trust model is to compute a score for each user $r \in \mathcal{R}$ using the trust network. The trust model estimates that $a$ will accept the answer proposed by the highest score member of $\mathcal{R}$. Local trust computes the score according to (10):

$$score(r) = \begin{cases} e(a,r) & if\ e(a,r) \in E \\ \sum_j \underline{\oplus}[e(a,f_j) \otimes e(f_j,r)] & elsewhere \end{cases} \quad (10)$$

where: $e(a,r)$ is the direct opinion (edge) of $a$ in $r$.
$f_j$ is a member of $F$, the set of the direct friends of $a$, formally: $f_j \in F : \Longleftrightarrow\ e(a,f_j) \in E$.
$\Sigma_{0 \le j \le N} \underline{\oplus}$ is the aggregation of multiple (exactly $N$) opinions. Note that $e(f_j,r)$ itself can be composed of the opinions of the friends of $f_j$.

To predict the accepted answer of a given question $q$ asked by the user $A$, we identify $\mathcal{R}$ the set of users who contributed answers to the current question. Then, we traverse the graph (trust network) to compute the local trust between person asking and each of them. We assume that $A$ will accept the answer of the most trustee user within $\mathcal{R}$. According to this model, $A$ consults his friends only about members of $\mathcal{R}$ with whom he has no direct interactions, otherwise considers only his own opinion. Consulted friends repeat the same strategy in consulting their friends. The drawback of this model is when $A$ has only one interaction with a member $r$ of $\mathcal{R}$, this might be not enough to evaluate him. $A$ may have a friend $B$ who has had many interactions with $r$ so more apt to evaluate $r$. According to this model $A$ will not ask $B$ about his opinion in $r$.

The aim of $A$ is to rank $\mathcal{R}$ by the trustworthiness of its members. Whenever he has no information about a member $r$ of $\mathcal{R}$, $A$ will ask his friends about their opinions in this very member. So the task of friends is to evaluate $r$ without any farther information. The more $A$ is connected, the faster is the

```
1: procedure INDIVIDUALTRUST(A, B)
2:     if (e(A, B) ∈ E) then
3:         return e(A, B)
4:     else
5:         e(A, B) ← e(0, 0, 1)                ▷ a neutral opinion
6:         for all f ∈ A.friends do
7:             e(A, B) ← e(A, B)⊕[e(A, B) ⊗ e(f, B)]
8:         end for
9:         return e(A, B)
10:    end if
11: end procedure
```

Fig. 3: Individual trust function

model, since the probability to have direct relationships with the members of $\mathcal{R}$ becomes higher. The pseudo code 3 shows how this model works in demanding friends' opinions.

### B. Collective trust

This model is based on collective opinions instead of personal opinions. In the previous model, collective opinions were used only in the case of absence of personal opinions. In this model, collective opinions are used in all cases. This semantically means that $A$ will ask his friends about all the members of $\mathcal{R}$, so even those who he already knows. Formally:

$$score(r) = \begin{cases} (a,r) \underline{\oplus} \sum\limits_{j} \underline{\oplus}[e(a,f_j) \otimes e(f_j,r)] \\ \qquad if\ e(a,r) \in E \\ \sum\limits_{j} \underline{\oplus}[e(a,f_j) \otimes e(f_j,r)] \\ \qquad elsewhere \end{cases} \tag{11}$$

This model assumes that direct interactions are frequently unable to assure sufficient information about users. In the previous model, user could supply a personal opinion about another user once he has at least one interaction with him. We think that this affects the quality of the opinion, because of the lack of experience. In the current model, user aggregates his opinion with the his friends' opinions, each friend's opinion is conditioned by the trust given to him by the active user. This means that we always need to traverse the graph, which can be time consuming in large graphs. We alleviate this problem by building a graph by domain in our data.

***Example:***

Back to the same example in Section II. Fig. 5 illustrates trust network extracted from the described relations in the example. So when $A$ asks a question to which she get replies from $E$, $F$ and $G$, then $\mathcal{R} = E, F, G$. $A$ needs to rank the members of $\mathcal{R}$ to identify the most trustworthy member.

For the individual trust model, scores are computed as follows:

$$score(E) = e(A,E)$$

$$score(F) = [e(A,B) \otimes e(B,F)] \underline{\oplus} [e(A,C) \otimes e(C,D) \otimes e(D,F)]$$

```
1: procedure COLLECTIVETRUST((A, R))
2:     Declare scores[R]
3:     for all score ∈ scores do score = e(0, 0, 1)        ▷ neutral opinion
4:     end for
5:     for all (r ∈ R do
6:         if opinion(A, r) ∈ E then
7:             scores[r] = e(A, r) ⊗ scores(r)
8:         end if
9:     end for
10:    for all f ∈ A.friends do
11:        fscore = collectiveTrust(f, R)
12:        for all r ∈ R do
13:            scores[r] = scores[r]⊕fscore[r]
14:        end for
15:    end for
16:    return scores
17: end procedure
```

Fig. 4: Collective trust function

$$score(G) = e(A,C) \otimes e(C,G)$$

As for the collective trust model, the scores of $F$ and $G$ do not change, but the score of $E$ becomes as follows:

$$score(E) = [e(A,E)] \underline{\oplus} [e(A,B) \otimes e(B,E)]$$

Now let us add a link between $C$ and $F$, and see the effect of such a link:

In individual trust model:

$$score(F) = [e(A,B) \otimes e(B,F)] \underline{\oplus} [e(A,C) \otimes e(C,F)]$$

In collective trust model:

$$score(F) = [e(A,B) \otimes e(B,F)] \underline{\oplus} [[e(A,C) \otimes e(C,F)] \underline{\oplus} [e(A,C) \otimes e(C,D) \otimes e(D,F)]]$$

Once again, we see that in individual trust model, when $A$ asks $C$ about his opinion in $F$, as $C$ has a direct link with $F$, he his response to $A$ is based only on this direct link. Whereas in collective trust model, for the same case, $C$ asks $D$ about this last's opinion about $F$, and return to $A$ the aggregation of the opinion $D$ conditioned by the trust between $C$ and $D$, and $C$'s own opinion.

### C. Global trust model (GTM)

Each question in stackexchange has a set of associated keywords. We use these keywords to build a new global trust model (GTM), that exploits the reputation of users towards keywords. When a user $A$ accepts the answer of a user $B$ to his question, a link is created or updated between $B$ and each of the keywords associated to the question, so we do not use neither a graph nor user to user connections. The semantic signification of the links between users and keywords is the experience of the user towards the keyword, so a reputed user towards a keyword can also be called expert. The profile of a user is represented by a hashtable where keys are the keywords

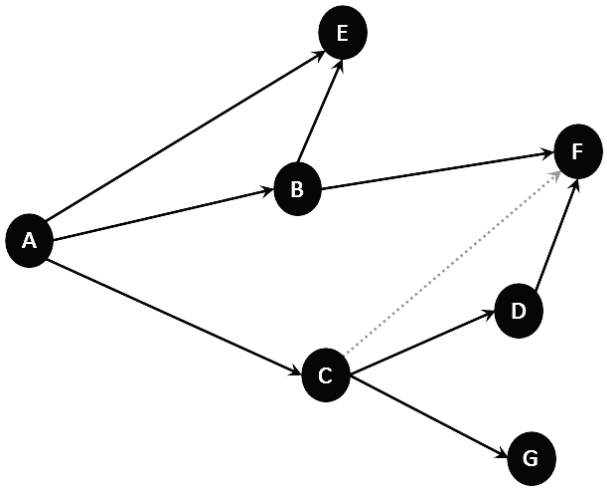


Fig. 5: Trust graph

and the values are subjective logic opinions to express his experience related to the keywords.

To predict the accepted answer of a given question $Q$ asked by the user $A$, we identify $\mathcal{R}$ the set of users who contributed answers to the current question, and the set $K$ of keywords associated to the question. We compute the average reputation score to each member of $\mathcal{R}$ towards the elements of $K$. The member with the highest average score is chosen to be the owner of the accepted answer.

In (LTM) and (CTM) only friends and their friends can influence the decision of the person asking, and their influence is limited by the trust that the person asking accord to each them. In the current model, all the users in the dataset can influence the reputation score of the members of $\mathcal{R}$ without conditions. This can affect the robustness of the model to malicious attacks.

## IV. Experimental work

We use the dataset of the website stackoverflow. The website offers a question answering forum for multiple domains, mainly but not limited to computer science. The available data contains 30 domains. Users subscribe to the website by domain, so one user can have multiple accounts, according to the number of domains in which he participates. The total number of accounts is 374,008 for about 153,000 users.

The user asks a question in a given domain, and associates a set of keywords to his question, then he receives many answers. He chooses the most relevant answer to him and attributes an "accepted answer" label to it. Nevertheless, users can keep proposing new answers. Subsequent users who have the same problem as the person asking can take advantage of the answers and rate them on their usefulness by attributing thumb-up or thumb-down. In the available dataset, we have access to only the total number of thumbs-up and the total number of thumbs-down an answer has, but no information about suppliers' identities. The website offers the possibility to order answers by relevance, where the accepted answer is put in the top of the list, followed by the other answers ordered by the difference between thumbs-up count and thumbs-down count. Our work aims to use trust based models to predict the accepted answer over the set of available answers. Total number of questions in current dataset equals to 371,594, for a total number of answers 816,487. We divide the questions of each domain in five equivalent sets. Then, we apply a crossing test in five iterations, in each iteration we use four sets for learning and building the trust network and the fifth for testing the prediction quality.

### A. Interpreting interactions

In stackoverflow, when a user $A$ asks a question, he receives a list of answers from many users. $A$ can accept only one answer. Unaccepted answers are not necessarily bad ones. They might be simply not good enough compared to the accepted one. They even might be better but arrived too late and $A$ has already accepted another satisfactory answer. Basically, while we do not have an explicit reaction from $A$ towards the unaccepted answers, we suppose four hypotheses to treat them:

1) rigorous hypothesis: unaccepted answers are considered as failed interactions.
2) ignoring hypothesis: unaccepted answers are not considered at all.
3) independent subjective hypothesis: in both previous methods, the interaction value is either +1 (successful), or -1 (failed). In this method, we introduce relatively successful/failed interactions. We use the rates of community towards the answer to estimate a subjective successful/failure of the interaction. In fact, the thumb-up represents a successful interaction with an unknown user, same thing for the thumb-down with a failed interaction. The global reaction of the community towards the answer is subjective opinion resulting from members' interactions with the answer. We consider the expectation value of the community's opinion as the value of the partially successful/failure of the interaction between the person asking and the replier.
4) dependent subjective hypothesis: regarding to the fact that a user can give a thumb-up for an answer because it is better/worse than others, the attribution of thumb-up and thumb-down can be relative too. The reason why we propose another subjective method where our certainty is influenced by the global number of thumb-up and thumb-down attributed to all answers of the same question. In this case, the opinion about an answer is dependent on the the other opinions about the other answers.

$$Certainty_j = \frac{\sum_j th}{2 + \sum_{i=an_0}^{an_n} \sum_i th}$$

where $th$ is an absolute value of thumb (up or down).
$j$ is the current answer.
$n$ is the number of answers of the current question.
The default non-informative prior weight W is normally defined as W = 2 because it produces a uniform Beta PDF in case of default base rate a = 1/2.
The three components of the opinion are:

$$belief_j = uncertainty_j \times \frac{\sum_j th_{up}}{\sum_j th}$$

where $\sum_j th_{up}$ is the number of thumbs up attributed to the answer.

$$disbelief_j = uncertainty_j \times \frac{\sum_j th_{down}}{\sum_j th}$$

where $\sum_j th_{down}$ is the number of thumbs down attributed to the answer.

$$uncertainty_j = 1 - certainty_j$$

Finally, we compute the expectation value of the resulting opinion and consider it as the value of the relative success/failure interaction.

## V. Evaluation

Our comparison includes three axes. The first one is the precision of prediction. The second is the complexity, which indicates the execution time of each model. The third is the robustness to malicious attacks.

### A. Precision

*Evaluation Metrics:* We consider the problem of finding the accepted answer as a list ranking problem with one relevant item. Mean reciprocal rank (MRR) is a quality metrics used to evaluate systems that have to give out a ranked list with only one relevant item. Reciprocal rank (RR) of question is $1/r$ where $r$ is the rank given by the evaluated algorithm to the accepted answer. Mean reciprocal rank is the mean value of RR's to all questions. The value of this metrics varies between 0 and 1, where 1 is the best precision score.

MRR is a good indicator to the performance of prediction algorithms for ranked lists. Nevertheless, we think that it is not perfectly adapted to our case. MRR is usually used for systems that have to predict a list of items within which a relevant item exists. We are trying to find the accepted answer by re-ranking an existing list of answers. Remark the case when the algorithm ranks the relevant item in the last position of the list, the algorithm is recompensed for at least having chosen the item within the list. In our case, the list is predefined, so the algorithm should not be recompensed for ranking the relevant item at the end of the list. The range of RR values is $[1/r, 1]$, we propose a modified version where the value varies between 1 if the relevant item is in the top of the list, and 0 if it is at the end of the list. We call this metrics mean predefined lists rank (MPLR), where predefined lists rank PLR is given by the formula:

$$PLR = \frac{N - r}{N - 1}$$

where: $N$ is the size of the list.

MPLR is the average of PLRs for all questions. We employ a modified competition ranking strategy, so the ranking gap is left before the *ex aequo* items. For example, if two items on the top of the list have the same score, they are considered both second, and no item is put at the top of the list.

TABLE II: MRR results

| method | MoleTrust | Local trust | Collective trust | Global trust |
|---|---|---|---|---|
| **Rigorous** | - | 0.57 | 0.88 | 0.884 |
| **Ignoring** | 0.53 | 0.58 | 0.75 | 0.7 |
| **Dependent probabilistic** | - | 0.62 | 0.87 | 0.815 |
| **Independent probabilistic** | - | 0.617 | 0.86 | 0.78 |

TABLE III: MPLR results

| method | MoleTrust | Local trust | Collective trust | Global trust |
|---|---|---|---|---|
| **Rigorous** | - | 0.37 | 0.85 | 0.85 |
| **Ignoring** | 0.3 | 0.36 | 0.69 | 0.6 |
| **Dependent probabilistic** | - | 0.442 | 0.84 | 0.76 |
| **Independent probabilistic** | - | 0.438 | 0.83 | 0.73 |

*Results and discussions:* Only questions with accepted answer and more than one proposed answer are appropriate for our test. The corpus contains 118,778 appropriate questions out of the 371,594 questions of the corpus.

As MoleTrust is not probabilist and does not consider the distrust, only the ignoring hypothesis is applicable on it. Table II illustrates the MRR scores of the four models, and Table III illustrates MPLR scores. MPLR scores are, of course, lower than those of MRR. Nevertheless, both tables lead to the same conclusions.

Obviously, all the SL models are more precise than MoleTrust, which guarantee certain improvement to the SL compared to the referential model.

Concerning the SL models, it is obvious that the precision of CTM and GTM surpass widely that of LTM.

Basically, the truster in LTM builds his belief by mainly exploiting his own interactions. Whereas, CTM leans fully on collective opinions that rely on more complete evidences than individual ones. Trustee friends enrich collective opinions by more knowledge, that make them more reliable and accurate. These results show the limit of individual opinions and local relationships, because direct interactions can be poorly informative, and relying only on them can lead to inaccurate decisions. A fellow in a social environment needs always to integrate and interact within communities to be more informed, and more capable to adjust his decisions.

GTM offers a larger archive of interactions to the trusters. Truster in GTM has access to all the past interactions of the trustee, so construct a more elaborated belief about him. The performance of GTM is largely better than LTM. On the other hand, it is less precise than CTM even though it makes use of more evidences. We assume that sometimes these supplementary evidences cause information overload, and tend to be noisier than profitable. In addition, GTM accord the same weight to the opinions of all participants, whether they were trustees or not form the active user perspectives.

We would refer to the difference in context consideration. LTM and CTM consider the domain of the question as a

context. GTM considers a more refined interpretation of the context, based on a sub-domain defined by the tags associated to the question. The context in GTM is very adaptive, this leads to a more specific person having competences in this exact context. The presence of this person in the list of people who answered the question proves his willingness to assist the person asking, his competence and mastery of subject lead him to be the owner of the accepted answer. For example, if $B$ was able to answer a question of $A$ about Java programming language, this does not mean that he would be able the next time to reply to a question about C++ programming language, although it is still the same domain (context) for LTM and CTM. So even within the same domain, people might be experts in narrow sub-domains, while having a general or even weak knowledge about the other parts of the domain. If $A$ tried to reply the question of $B$ about C++, only GTM will detect that he is not the best person to reply in the domain of "c++ programming", whereas LTM and CTM will consider him a good candidate because he is trustee in the domain of "programming". Current precision score do not allow to evidently evaluate the influence of both consideration.

In real life, regret can assist to re-establish trust. The structure of local trust systems does not possess any mechanism to reconsider relationship after a bad integration with a destination user (which can be occasional), collective opinions allow the reconsideration of the relation with this user if he was trustee by intermediate friends of source user.

Regarding the four hypotheses about treating unaccepted answers in LTM, we find that probabilistic methods are slightly better than both rigorous and ignoring hypotheses. In CTM and GTM, the three hypotheses that try to infer from unaccepted answers surpass the performance of the forth that neglects these information (ignoring hypothesis). We conclude that unaccepted answers can be profitable, and then should not be neglected. Extracting information from these answers is possible thanks to the flexibility of subjective logic. This framework proves again its capability to deal with incomplete evidence cases.

### B. Complexity

Complexity is an important issue to evaluate algorithms. The importance of complexity evaluation is to estimate the time needed for each model to be executed. A good recommender must be able to generate recommendation in a reasonable delay.

Algorithm complexity is a function of $t(n)$, where $n$ is the input size. The complexity function gives a clue about the expected execution time of the algorithm given an input of size $n$. Complexity calculus is independent from the hardware, the programming language, the compiler and the implementation details. It takes in consideration only the elementary operations of the algorithm such as: variable assignment $(t(n) = 1)$, comparison $(t(n) = 1)$, loop on a list of size $n$ $(t(n) = n)$, comparing all the values of an array to each other $(t(n) = n^2)$, traversing a graph ($t(n) = V + E$), where $V$ is the number of vertices, and $E$ is the number of edges).

The big O notation is used to refer to the complexity, this notation keeps only the elementary element that maximize the algorithm complexity. For example, having an algorithm with ($t(n) = n^2 + 4 \cdot n + 2$), the equivalent in big O notation is $O(n) = n^2$.

Generally, the evaluation of complexity takes into account the worst case and the average case. The worst case represents the upper bound of time needed to execute the algorithm, and the average case is the lower bound.

Graph traversal complexity equals to $O(V + E)$. In the worst case, MoleTrust, LTM and CTM have to execute this operation $R$ times, where $R$ is the number of users who have proposed answers to the question. By consequence, the complexity of these three models equals to $O(R \cdot (V + E))$. The complexity of the GTM is $O(R \cdot L)$, where $L$ is the size of the list of keywords with which the member of $R$ has a reputation score.

In the worst case, MoleTrust, LTM and CTM have the same complexity. We can consider that the GTM is less complex whereas $L$ is usually smaller than $V + E$.

As the worst case is mostly infrequent, it is usually accompanied by the average case complexity. We define $R'$ as the subset of $R$ that contains the users having no direct trust relationship with the active user, so $R' \subseteq R$. The average complexity of LTM is $O(R' \cdot (V + E))$. It is obvious that average complexity of CTM is the same as its worst case complexity. The average complexity MoleTrust is less than LTM and CTM, because it stops searching when it finds the first member of $R$. Basically the average complexity of the GTM equals also to $O(R \cdot L)$ when using lists. The average complexity of hashtables is $O(1)$.

Finally, from the perspective of complexity we find that GTM is the less complex, followed by LTM, and CTM is the most complex one, so the most time consuming. This complexity analyses illustrates the limitation of CTM for the applications with huge graphs.

### C. Robustness against malicious attacks

In a malicious group attack scenario, we distinguish three groups of users. The attackers who participate in the execution of the attack. The affected users whose recommendations are contaminated because of the attack. And the pure users who are untouched by the attack.

In the group attack many profiles cooperate to achieve the attack's goal. These profiles can be possessed by one or more user, they unite to improve the score of one or more of them to a point that they can control the recommendations generated to other users. In the current application a group of profiles might ally together to execute a group attack. The members of group keep mutually inserting questions, answering them, and accepting each others' answers. While the application is contextualized, and the trust models treat the domains separately, attackers must target a given domain or repeat the same operation for each domain.

GTM is weak to this kind of attacks. The group can augment the reputation score of its members for chosen keywords, and contaminate them. Hence, when any pure user asks a question containing contaminated keywords, he will become affected and receive a contaminated recommendation from the attackers.

In MoleTrust, the local and the collective models, the topology of the graph assists to isolate the group of attackers. The communitarian behaviour will make them highly connected to each other but weakly connected to other users. Hence, a pure user can not be affected unless he decides himself to trust one or more attackers, which is very unlikely. Even if this happens once by accident, the resulting link is not strong enough (especially in CTM), because it is based on one interaction, and it will be more uncertain than other links, so with weak influence.

In [36], the authors propose the bottleneck property to state about the robustness of a trust model to the group attack. The meaning of the bottleneck property is that when having a trust relation $s \rightarrow t$, where $s$ is a pure user and $t$ is an attacker, this relation is not significantly affected by the successors of $t$. Fig. 6 illustrates an attacked graph with a bottleneck property.

The edges in our models are formed of SL opinions. So the only way to strengthen this relation, is by more successful interactions between $s$ and $t$, which is decided by $s$ himself. To summarize, in local and collective model, the attack can succeed only when pure users decide deliberately to trust attackers.

The conclusion of this analysis is that the global model is weaker than the local and the collective models against malicious group attacks.

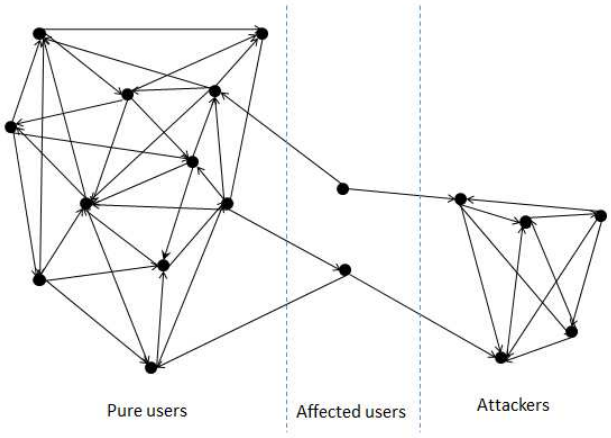


Fig. 6: The bottleneck phenomena in the trust graph

## VI. Conclusion and Future Works

In this paper, we compared three different interpretations of computational trust model.

We effected a comparison that consists of three axes (precision, complexity, and robustness). Fig. 7 resumes the conclusions. In terms of precision, we showed the limits of individual opinions compared to collective and global ones. Using opinions based on evidences from multiple resources is more fruitful, with some reservations to information overload limits. We represent that in Fig. 7, by putting CTM and GTM closer to the precision circle than LTM.

Although CTM has the best precision score, it still the most complex model among the three studied model. In Fig. 7, GTM

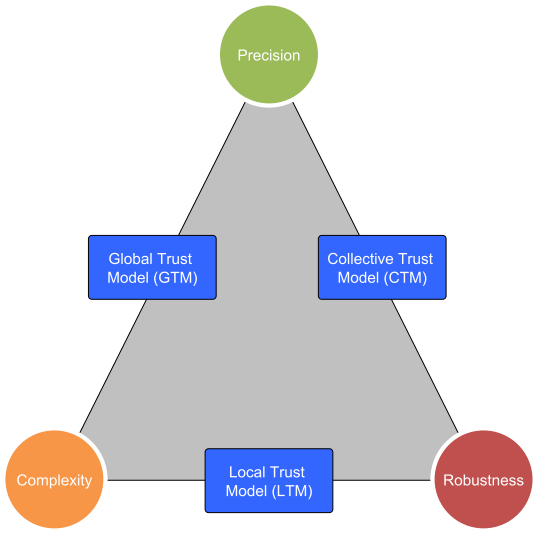


Fig. 7: The triple evaluation of the three trust models

and LTM are closer to the circle of complexity, because they have a better (lower) complexity. Even though GTM is less complex than LTM.

GTM forms a compromise between precision and complexity. Yet, its weak point is in the robustness axe. It is theoretically weaker than the other two models. In Fig. 7, it is located far from the robustness circle.

Our study puts the light on a weak point of each model. So the choice of a model is still dependant on the type application, the context and the desired characteristics.

Some of our results are theoretically inferred (the robustness issue). We are interested in proving that empirically, by simulating malicious attacks on the dataset, in order to measure the influence of these attacks on the precision of each model.

## References

[1] C. Haydar, A. Roussanaly, A. Boyer *et al.*, "Individual opinions versus collective opinions in trust modelling," in *SOTICS 2013, The Third International Conference on Social Eco-Informatics*, 2013, pp. 92–99.

[2] C. Haydar, A. Boyer, and A. Roussanaly, "Local trust versus global trust networks in subjective logic," in *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, vol. 1. IEEE, 2013, pp. 29–36.

[3] M. Richardson, R. Agrawal, and P. Domingos, "Trust management for the semantic web," in *In proceedings of the second international semantic web conference*, 2003, p. 351368.

[4] P. Massa and B. Bhattacharjee, "Using trust in recommender systems: an experimental analysis," *Trust Management*, p. 221235, 2004.

[5] R. Falcone, G. Pezzulo, and C. Castelfranchi, "A fuzzy approach to a belief-based trust computation," in *Lecture Notes on Artificial Intelligence*. Springer-Verlag, 2003, p. 7386.

[6] F. Almenrez, A. Marn, C. Campo, and C. G. R, "PTM: a pervasive trust management model for dynamic open environments," in *First workshop on pervasive security, privacy and trust PST04 in conjuntion with mobiquitous*, 2004.

[7] W. Tang, Y.-X. Ma, and Z. Chen, "Managing trust in peer-to-peer networks," *Journal of Digital Information Management*, vol. 3, no. 2, p. 58, Jun. 2005.

[8] I. Zaihrayeu, P. P. D. Silva, D. L. Mcguinness, I. Zaihrayeu, P. Pinheiro, S. Deborah, and L. Mcguinness, "IWTrust: improving user trust in answers from the web," in *Proceedings of 3rd International Conference on Trust Management (iTrust2005*. Springer, 2005, p. 384392.

[9] J. O'Donovan and B. Smyth, "Trust in recommender systems," in *Proceedings of the 10th international conference on Intelligent user interfaces*, 2005, p. 167174.

[10] P. Victor, C. Cornelis, M. D. Cock, and P. P. D. S. B, "Gradual trust and distrust in recommender systems. fuzzy sets and systems," in *In press*, 2009.

[11] M. De Cock and P. P. da Silva, "A many valued representation and propagation of trust and distrust," in *Proceedings of the 6th international conference on Fuzzy Logic and Applications*, ser. WILF'05. Berlin, Heidelberg: Springer-Verlag, 2006, p. 114120.

[12] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust," in *Proceedings of the 13th international conference on World Wide Web*, 2004, p. 403412.

[13] U. Kuter and J. Golbeck, "Using probabilistic confidence models for trust inference in web-based social networks," *ACM Trans. Internet Technol.*, vol. 10, no. 2, p. 8:18:23, 2010.

[14] B. Mobasher, R. Burk, R. Bhaumik, and C. Williams, "Toward trustworthy recommender systems an analysis of attack models and algorithm robustness.pdf," vol. 7, no. 4, 2007.

[15] R. Burke, B. Mobasher, R. Zabicki, and R. Bhaumik, "Identifying attack models for secure recommendation," in *Beyond Personalization: A Workshop on the Next Generation of Recommender Systems*, 2005.

[16] (2013, Sep.) Stack overflow website (data collected in sep 2011). [Online]. Available: http://stackexchange.com/

[17] A. Josang. (2013, Sep.) Subjective logic. [Online]. Available: http://folk.uio.no/josang/papers/subjective_logic.pdf

[18] M. Kwan and D. Ramachandran, "Trust and online reputation systems," in *Computing with Social Trust*, ser. HumanComputer Interaction Series, J. Golbeck, Ed. Springer London, Jan. 2009, pp. 287–311.

[19] K. Krukow, *Towards a Theory of Trust for the Global Ubiquitous Computer: A Dissertation Presented to the Faculty of Science of the University of Aarhus in Partial Fulfilment of the Requirements for the PhD Degree*. Department of Computer Science, University of Aarhus, 2006.

[20] C. N. Ziegler and G. Lausen, "Spreading activation models for trust propagation," in *e-Technology, e-Commerce and e-Service, 2004. EEE'04. 2004 IEEE International Conference on*, 2004, p. 8397.

[21] A. Abdul-Rahman and S. Hailes, "Supporting trust in virtual communities," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000*, Jan. 2000, p. 9 pp. vol.1.

[22] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in P2P networks," in *Proceedings of the 12th international conference on World Wide Web*, ser. WWW '03. New York, NY, USA: ACM, 2003, p. 640651.

[23] R. R. Yager, J. Kacprzyk, and M. Fedrizzi, Eds., *Advances in the Dempster-Shafer theory of evidence*. New York, NY, USA: John Wiley & Sons, Inc., 1994.

[24] A. Dempster, "The DempsterShafer calculus for statisticians," *International Journal of Approximate Reasoning*, vol. 48, no. 2, pp. 365–377, 2008.

[25] J. A. Golbeck, "Computing and applying trust in web-based social networks," Ph.D. dissertation, University of Maryland at College Park, College Park, MD, USA, 2005, AAI3178583.

[26] L. Mui, *Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks*. PhD Thesis, Massachusetts Institute of Technology, 2002.

[27] A. Josang, "A logic for uncertain probabilities," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, no. 03, p. 279311, 2001.

[28] A. Josang, R. Hayward, and S. Pope, "Trust network analysis with subjective logic," in *Proceedings of the 29th Australasian Computer Science Conference-Volume 48*, 2006, p. 8594.

[29] A. Josang and S. Pope, "Semantic constraints for trust transitivity," in *Proceedings of the 2nd Asia-Pacific conference on Conceptual modelling-Volume 43*, 2005, p. 5968.

[30] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, 1999.

[31] S. D. Ramchurn, D. Huynh, and N. R. Jennings, "Trust in multi-agent systems," vol. 19, p. 2004.

[32] K. McNally, M. P. O'Mahony, B. Smyth, M. Coyle, and P. Briggs, "Towards a reputation-based model of social web search," in *Proceedings of the 15th International Conference on Intelligent User Interfaces*, ser. IUI '10. ACM, p. 179188.

[33] A. Jøsang and J. Golbeck, "Challenges for robust trust and reputation systems," in *Proceedings of the 5th International Workshop on Security and Trust Management (SMT 2009), Saint Malo, France*, 2009.

[34] T. Zuva, S. O.Ojo, S. Ngwira, and K. Zuva, "A survey of recommender systems techniques, challenges and evaluation metrics," *International Journal of Emerging Technology and Advanced Engineering*, 2012.

[35] P. Massa and P. Avesani, "Trust-aware collaborative filtering for recommender systems," *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, p. 492508, 2004.

[36] R. Levien, "Attack resistant trust metrics," Tech. Rep., 2004.