



HAL
open science

Characterizing and using gullibility, competence, and reciprocity in a very fast and robust trust and distrust inference algorithm for weighted signed social networks

Karim Akilal, Mawloud Omar, Hachem Slimani

► To cite this version:

Karim Akilal, Mawloud Omar, Hachem Slimani. Characterizing and using gullibility, competence, and reciprocity in a very fast and robust trust and distrust inference algorithm for weighted signed social networks. Knowledge-Based Systems, 2020. hal-03034580

HAL Id: hal-03034580

<https://hal.science/hal-03034580>

Submitted on 1 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Characterizing and using gullibility, competence, and reciprocity in a very fast and robust trust and distrust inference algorithm for weighted signed social networks

Karim Akilal^{a,*}, Mawloud Omar^b, Hachem Slimani^a,

^aLaboratoire d'Informatique Médicale, Faculté des Sciences Exactes, Université de Bejaia, 06000 Bejaia, Algérie

^bUnité de Recherche LAMOS, Faculté des Sciences Exactes, Université de Bejaia, 06000 Bejaia, Algérie

Abstract

Predicting trust is a classic problem in social networks analysis. Furthermore, while most early approaches ignore distrust, recent works seem to consider it as important, if not more important, than trust itself. In this paper, we present a novel approach to predict both trust *and* distrust in Weighted Signed Social Networks very efficiently and in a satisfyingly accurate and robust way. Therefore allowing people to have healthier online presence and interactions.

Being a local metric that does not rely on trust propagation, the proposed approach does not suffer from some serious limitations like trust decay, opinions conflict, path dependence, and time complexity. Moreover, our experiments on four real-world datasets show that, in addition to its simplicity and extensibility, this algorithm is robust to network sparsity, and provides satisfyingly accurate and very fast predictions.

Keywords: Online social network, Trust inference, Distrust, Trust metric, Social trait

1. Introduction

1.1. Motivation

Let us face it, the open nature of online social networks (OSNs) is a double-edged sword. Indeed, billions of people share knowledge and *socialize* thanks to this nature. Yet, because of this openness, malicious users are also able to spread misinformation and, to some extent, harm others (Shneiderman, 2015). People should think twice before believing, downloading, or sharing something online. They should assess the trust they are willing to put in others, for, to quote Robbins (2016), “*Trust matters*”. It always did, and perhaps even more so today where *friendship* is claimed with a click on a button. The user, seemingly surrounded by thousands of friends, is sorely alone when it comes to decide what to believe and whom to trust. In such a world where information flows

*Corresponding author

Email addresses: karim@akilal.com (Karim Akilal), mawloud.omar@gmail.com (Mawloud Omar), haslimani@gmail.com (Hachem Slimani)

quickly and incessantly, we, therefore, need tools that are able to assist users in deciding how much should they trust (or distrust) others.

Often, the problem that these solutions aim to solve is stated in a single question: “How much should a user X (trustor) trust or distrust another user Y (trustee)?” As simple as it may seem, this question carries with it at least another question: “What is distrust?”

While scholars seem to agree on the meaning of trust (Rousseau et al., 1998), distrust fails to bring them to a consensus (Guo et al., 2017). As noted by Kim and Ahmad (2013); and Ziegler (2013), most early trust prediction approaches simply ignore distrust or consider it as the absence of trust. However, as argued by Hawley (2013) and others, distrust is *not* the mere absence of trust. Distrust means pessimism, and even fear (Lewicki et al., 1998). It calls for vigilance and precautions whereas neutrality, in this context, may be assimilated to indifference. These precautions that have to be taken are the reason why authors like DuBois et al. (2011) rightly suggest that knowing whom to distrust is equally, if not more, important than knowing whom to trust.

1.2. Challenges

Unfortunately, adapting most trust-only approaches to consider distrust is impractical and often impossible (Guha et al., 2004; Chiang et al., 2014; Tang et al., 2016). For instance, algorithms that rely on the transitivity aspect of trust fail to properly handle distrust, because distrust is *not* transitive, as empirically shown by Tang et al. (2014) and later by Gao et al. (2016).

In addition to handling distrust, trust prediction algorithms have to address other important challenges that we summarize in the following criteria:

1. **Accuracy** : When a user X has to decide if he/she can safely share something with another user Y , or to believe what Y shares, the algorithm should provide a good estimation of Y 's trustworthiness. The more accurate this estimation is, the easier and the more adequate would X 's decision be.
2. **Robustness** : Prediction often relies on prior knowledge —*The more the merrier*. Predicting trust relations depends on other previously known trust relations in the network. However, some (or most) of these relations may be hidden (for privacy concerns) or simply unavailable (because of technical difficulties). A trust prediction algorithm should perform equally well, even when prior knowledge is scarce. Simply stated, robustness to network sparsity is the ability of an algorithm to still make acceptably accurate predictions despite the scarcity of prior knowledge.
3. **Speed** : Social networks users are *flooded* with information to consume, and with events that require a quick action. A mistake is often a click away. Trust prediction algorithms should be as quick as possible to provide a prediction, because a false belief is often hard to correct, and a sent message is hard to recall. Indeed, preventing a user from downloading a malware, from taking a medical advice from an impostor, or from sending sensitive information to untrustworthy recipients, requires trust recommendations that are as fast as possible.

Of course, these criteria are hard to satisfy at once. To be accurate, algorithms might need to make intricate computations, therefore be slow to return a prediction. They might also require several known trust relations to accurately predict an unknown one,

hence provide poor predictions when there is less known relations to infer from. Put simply, accuracy conflicts with speed and robustness.

We believe that the best answer to this dilemma is an approach that —by design— *depends* on as less prior knowledge as possible —an algorithm that would need the bare minimum of known trust relations to function. Such an algorithm, if able to provide accurate predictions, will also be fast and robust to networks sparsity.

Our proposed approach, which subscribes to this design, is based on localized nodes metrics. It consists in expressing and using metrics that describe social traits that affect the way we *give* or *receive* trust. Among these traits are gullibility, competence, and reciprocity. They, definitely, influence how we usually trust (or are trusted).

These traits can be simply computed without having to perform any graph traversal; because they only depend on the direct neighbors of the involved actors in a trust relation (the trustor and the trustee). Our main hypothesis is that the most prevailing trait would have more influence on the act of trust. We thus think of these traits as forces competing to affect trust. And, indeed, the satisfying results from our experiments on four real-world datasets show that this hypothesis stands.

In addition to its simplicity, the proposed approach is not subject to the limitations of propagative approaches such as:

Trust decay: When propagated along lengthy paths, trust indeed decays (Liu et al., 2011), and that is quite natural. In fact, most propagative approaches tend to limit their propagation horizons to a few hops, not only because of efficiency concerns, but because of accuracy that decreases beyond some hops (Golbeck, 2005a; Ziegler and Golbeck, 2015).

Path dependence: When inferring trust using transitivity, there might be multiple paths from a source to a sink that share some segments. This is problematic in the sense that some individuals are consulted twice (or more) (Jiang et al., 2016a). For example, say that we are trying to predict how much would node s trust another one k in a trust graph with the following paths: (s, u, k) , (s, v, k) , and (s, v, u, k) . The problem, in this case, is were we to explore the three possible paths from s to k , then we would consider u 's opinion twice, and v 's opinion just once.

Opinion conflict: One of the main aspects of trust is its subjectivity. People often have different opinions regarding others (Massa and Avesani, 2005). Therefore, when trying to predict trust using transitivity, one might encounter some conflicts of opinions (Jøsang and Pope, 2005). Which opinions should we favor in these cases? Resolving opinion conflicts is still an open problem in research, and according to Jiang et al. (2016a), research in other fields, such as sociology, can be introduced to solve this challenge.

Time complexity: Trust prediction using path discovery is time-consuming. In fact, as argued by Ghavipour and Meybodi (2018), using all paths in trust inference becomes impractical as social networks are usually massive in size.

1.3. Highlights of the proposed approach

The proposed approach aims to avoid these limitations by using social traits instead of transitivity. Indeed, rather than propagating trust along paths in a trust graph, our

strategy is to try to understand what affects trust by exploring some social traits of the protagonists in a trust relation (the trustor and the trustee). As suggested by Jiang et al. (2016a), one way to solve opinion conflicts in trust inference is to fully understand personal biases and features of the trustor. The present work adheres to this vision, and extends it to trustees as well. More specifically, we believe that knowing *everything* that makes people trust (or be trusted) should allow us to predict unknown trust values more accurately.

Some of the advantages of the proposed approach may be summarized as follows:

- Ability to predict both trust and distrust, contrary to most prediction algorithms in the literature.
- Novelty: because it does not rely on the transitivity of trust and its limitations.
- High efficiency: since it operates on the very direct neighbors of the trustor and the trustee.
- Robustness to network sparsity: since it uses minimal set of prior knowledge for inference.
- intuitivity: it is easy to understand and extend.
- Simplicity: it is easy to implement since it boils down to calculating euclidean distances between vectors.

We believe that these properties of the proposed approach make it a very suitable solution for trust (and distrust) prediction in social networks, thus assisting users in known whom (and how much) to trust or distrust. Such a knowledge will ultimately allows these individuals to have a healthier online presence by taking advantage of the positive aspects of social networks, and avoiding their known drawbacks.

The rest of this paper is organized as follows. In Section 2, we give a brief review of some related work. In Section 3, we start by giving definitions and mathematical representations of the three social traits (gullibility, competence, and reciprocity), then we present a simple algorithm that uses these traits to predict trust. Next, to validate our approach, we conduct, in Section 4, some experiments to evaluate the performances, the efficiency, and the robustness of the proposed approach. We discuss the results of these experiments in Section 5, and conclude this paper in Section 6 with a summary and some perspectives of future work.

2. Related work

Several approaches have been proposed to predict trust in social networks, some focusing on trust only, others on both trust and distrust. These approaches also differ in the way they work. According to Tang and Liu (2015), some are supervised (using machine learning techniques), others are unsupervised and need no prior training. Some are supported by additional information such as interaction data among users (Kim and Ahmad, 2013; Huang et al., 2018), interest similarity Mao et al. (2019), or emotions (Beigi et al., 2016). Others operate on the sole trust graph with no additional information. Some are probabilistic (Kuter and Golbeck, 2010), others are subjective logic inspired (Jøsang and

Pope, 2005), etc. For the sake of brevity, we refer the reader to, some excellent surveys such as those by Ruan and Durrezi (2016); Jiang et al. (2016a); Tang et al. (2016); and we focus hereafter on some unsupervised graph-based methods that do not require any interaction data. These approaches operate on two types of metrics: local and global ones. Local metrics describe how would a node u trust another one v . Global metrics, for their part, answer how trustworthy, or leaning to trust, is a given node (Massa and Avesani, 2007; Tang and Liu, 2015). These families of approaches have their merits and their flaws but they ultimately aim to answer, albeit differently, the same question: “How much should a user X trust another user Y ”.

2.1. Local metrics trust prediction

Most local metrics approaches use trust propagation rules that were enumerated by Guha et al. (2004). The most common of these rules is trust transitivity, which states that if a node u trusts another one v which itself trusts a third one w , then u may trust w to some extent (Golbeck, 2005b). Among these algorithms we can cite: TidalTrust (Golbeck, 2005a), MoleTrust (Massa and Avesani, 2007), SWTrust (Jiang et al., 2014), GFTrust (Jiang et al., 2016b), DLATrust (Ghavipour and Meybodi, 2018), and many others. These algorithms propagate trust through a trust graph from a source u to a sink v . Results from different paths are then aggregated to obtain a final value that would be an estimation of how much u would trust v . Applying these approaches *as-is* on signed networks (those with distrust relations) is, however, not straightforward. Indeed, empirical evidence by Gao et al. (2016) has shown that distrust is *not* transitive.

Still, distrust should not be dismissed. The need to predict distrust, and the advantages that negative links bring to social networks analysis (Papaoikonomou et al., 2013; Kunegis et al., 2013) make these challenges worth addressing. In fact, many efforts were taken in this direction. For instance, in Appleaseed (Ziegler, 2013), a propagative algorithm inspired by *spreading activation models*, trust is considered as an energy passing from nodes to their trustees, and distrust is modeled as a negative energy. Another interesting propagative approach was proposed by Gao et al. (2016). In their STAR algorithm, the authors defined a semiring (an algebraic structure) that operates on 2D values: trust (or distrust) and certainty. This semiring favors arcs with bigger certainty values and circumvents the intransitivity of distrust by simply ignoring paths with two successive negative links. Taking a different approach that does not rely on trust transitivity, Akilal et al. (2019) proposed a collaborative filtering based algorithm using agreement as a similarity metric to infer both trust and distrust relations by using only information from the direct neighbors of the trustors and the trustees. In addition to their time complexity (Ghavipour and Meybodi, 2018), and as noted by Jiang et al. (2016a), propagative approaches suffer from path dependence, trust decay, and opinion conflict.

2.2. Global metrics trust prediction

Several global metrics approaches were proposed in the literature. For example, the PageRank (Page et al., 1999) and the HITS (Kleinberg, 1999) algorithms that were initially proposed to rank web pages, have been adapted to social networks (Hu et al., 2018; Zhao et al., 2018). And the EigenTrust algorithm by Kamvar et al. (2003) has also known many variants (Chiluka et al., 2012; Kurdi, 2015). These metrics are designed for unsigned networks, and thus ignore negative links.

As for signed networks, Zolfaghar and Aghaie (2010) proposed some metrics like the popularity and the gregariousness of a node. While taking into account links signs, these metrics do not consider their weights. Later, Mishra and Bhattacharya (2011) introduced BIAS and DESERVE. These two global metrics respectively describe the bias of a trustor, and the prestige of a trustee. Computed iteratively, they describe more or less how trust is given and received. The PageRank and the HITS algorithms were also revisited to take into account negative links (de Kerchove and Dooren, 2008; Shahriari and Jalili, 2014). More recently, and in the spirit of what Mishra and Bhattacharya (2011) have proposed, Kumar et al. (2016) defined two new global metrics: FAIRNESS and GOODNESS. The authors state that we can infer how much should a node u (dis)trust another one v , by multiplying the fairness of u by the goodness of v .

The most common argument against global metrics, as argued by Massa and Avesani (2005) and Tang and Liu (2015), is that trust is subjective. As such, global metrics cannot describe how would a *specific* node u trust another one v , for v may be controversial and thus likely to be trusted differently by different nodes of the network.

The limitations of the propagative and the global-metrics-oriented approaches inspired us to propose a different approach that consists in expressing some *localized node metrics* that affect trust. Taken separately, these metrics are often of little use, but when involved as forces that influence the *act* of trust, they allow us to design a very simple algorithm that is quite reliable, very fast, and robust to network sparsity.

Naturally, there are many social traits that affect trust such as popularity, social conformity (Li et al., 2018), confirmation bias (Zollo and Quattrociochi, 2018), and many other user biases (Pavleska and Blažič, 2016; Braga et al., 2018). For the time being, our work focuses on three traits: the gullibility, the competence, and the reciprocity of individuals in a social network. However, as we will see in the next section, the proposed trust prediction algorithm is easily extensible to more traits. The considered traits and our prediction algorithm are described in more details in the next section.

3. Our proposed approach

As stated earlier, our approach does not rely on trust propagation. Instead, it consists in characterizing some social traits that affect trust, and then confronting them in a tug-of-war like game to predict trust (and distrust).

Although there are many social traits that affect trust, we consider in this study only three of them. This choice was mainly based on their simplicity and ease of comprehension (as the reader may see in what follows). We, nonetheless, consider adding more traits to the tug-of-war game in a future work.

3.1. Notation and preliminaries

We represent a social network by a weighted directed graph, where the nodes of the graph are individuals of the network, and the arcs are trust relations between said individuals. A trust relation from a node u (trustor) to another node v (trustee) is weighted using a real value $t(u, v)$ in the interval $I = [m, \mathfrak{M}]$ (where $m < 0$ and $\mathfrak{M} > 0$), such that the more u trusts v , the more $t(u, v)$ is positive, and the more u distrusts v , the more $t(u, v)$ is negative. Table 1 summarizes the adopted notation.

Notation	Meaning
$\mathcal{G}(\mathcal{N}, \mathcal{E}, t)$	A weighted directed graph \mathcal{G} with nodes in \mathcal{N} connected by arcs in \mathcal{E} that are weighted using the mapping t .
$t(u, v)$	Weight of the arc going from node u to node v .
$\overrightarrow{\Gamma}(u)$	Set of the trustees of the node u .
$\overleftarrow{\Gamma}(v)$	Set of the trustors of the node v .
$m < 0$	Minimal value of trust (extreme distrust).
$\mathfrak{M} > 0$	Maximal value of trust (extreme trust).
I	Trust interval, that is $[m, \mathfrak{M}]$.
\mathcal{R}	Trust range. $\mathcal{R} = \mathfrak{M} - m$.
\overrightarrow{m}_k	The vector (m, m, \dots, m) in \mathbb{R}^k of which all k components are equal to m .
$\overrightarrow{\mathfrak{M}}_\ell$	The vector $(\mathfrak{M}, \mathfrak{M}, \dots, \mathfrak{M})$ in \mathbb{R}^ℓ of which all ℓ components are equal to \mathfrak{M} .
$d(\mathbf{u}, \mathbf{v})$	The Euclidean distance between vectors \mathbf{u} and \mathbf{v} .
$g^+(u)$	Positive gullibility (or gullibility for short) of the node u .
$g^-(u)$	Negative gullibility (or paranoia) of the node u .
$c^+(u)$	Positive competence (or competence for short) of the node u .
$c^-(u)$	Negative competence (or incompetence) of the node u .
$\tau(u)$	Reciprocity of the node u .

Table 1. Notation used throughout this paper.

We use Iverson Brackets (Knuth, 1992; Graham et al., 1994). This notation makes an integer (0 or 1) from a logical statement P put between brackets as follows ¹:

$$[P] = \begin{cases} 1 & \text{if } P \text{ is true,} \\ 0 & \text{if } P \text{ is false.} \end{cases}$$

3.2. Problem definition

Let $\mathcal{G} = (\mathcal{N}, \mathcal{E}, t)$ be a directed and weighted graph representing a social network, where \mathcal{N} is the set of nodes, \mathcal{E} the set of arcs between nodes of \mathcal{N} , and $t: \mathcal{E} \mapsto [m, \mathfrak{M}]$ a mapping that associates to each arc (u, v) a weight $t(u, v)$ that represents the trust ($t(u, v) > 0$) or distrust ($t(u, v) < 0$) that the node u puts in the node v . Our work aims to quickly predict how much would a node u (dis)trust another node v . And that when all, or only some, other trust values are known. To simplify our narrative, we will use *trust* to denote both states: trust (positive values) and distrust (negative values).

3.3. Localized metrics based on social traits

Before diving into the details of our algorithm, we define hereafter three metrics that describe some social traits of nodes; namely, the gullibility, the competence, and the propensity to reciprocate trust.

¹Note that we adopt the “strong zero” convention described by Knuth as follows: “In general, when an Iverson-bracketed statement is false, we want it to evaluate into a “very strong 0,” namely a zero so strong that it annihilates anything it is multiplied by—even if that other factor is undefined.” (Knuth, 1992).

Definition 3.1. We say that a node u is *absolutely gullible* if it extremely trusts all of its trustees. i.e., $\forall v \in \vec{\Gamma}(u), t(u, v) = \mathfrak{M}$. Similarly, we say that a node u is *absolutely paranoid*, if it extremely distrusts all of its trustees. i.e., $\forall v \in \vec{\Gamma}(u), t(u, v) = \mathfrak{m}$.

These extreme cases are unlikely to exist. We use them, however, as references for estimating the *gullibility* and the *paranoia* of a node u as follows:

3.3.1. Gullibility

Let $u \in \mathcal{N}$ be a node, and let $\vec{\mathbf{T}}(u) = (t(u, v_1), t(u, v_2), \dots, t(u, v_n))$ be a vector in I^n comprised of the trust values that u puts in its trustees $v_i \in \vec{\Gamma}(u), i = 1, \dots, n$. The *gullibility* of the node u is given by the function $g^+ : \mathcal{N} \mapsto [0, +1]$ defined as follows:

$$g^+(u) = \left[\vec{\Gamma}(u) \neq \emptyset \right] \left(1 - \frac{d(\vec{\mathbf{T}}(u), \vec{\mathfrak{M}}_n)}{\mathcal{R}\sqrt{n}} \right), \quad \text{where } n = |\vec{\Gamma}(u)|. \quad (1)$$

3.3.2. Paranoia

Similarly, we define the *paranoia* of a node $u \in \mathcal{N}$, using the function $g^- : \mathcal{N} \mapsto [0, +1]$ that is defined as follows.

$$g^-(u) = \left[\vec{\Gamma}(u) \neq \emptyset \right] \left(1 - \frac{d(\vec{\mathbf{T}}(u), \vec{\mathfrak{m}}_n)}{\mathcal{R}\sqrt{n}} \right), \quad \text{where } n = |\vec{\Gamma}(u)|. \quad (2)$$

The idea behind Eq (1) and Eq (2) is illustrated in Fig. 1. Simply put, the closer $\vec{\mathbf{T}}(u)$ is to $\vec{\mathfrak{M}}_n$, the more gullible u is. And the closer $\vec{\mathbf{T}}(u)$ is to $\vec{\mathfrak{m}}_n$, the more paranoid u is.

Remark 1. In order to make $g^+(u)$ and $g^-(u)$ values in $[0, +1]$, the distances in Equations (1) and (2) are divided by $\mathcal{R}\sqrt{n}$ (which is the biggest distance between two points in a n -dimensional hypercube with an edge length equal to \mathcal{R}).

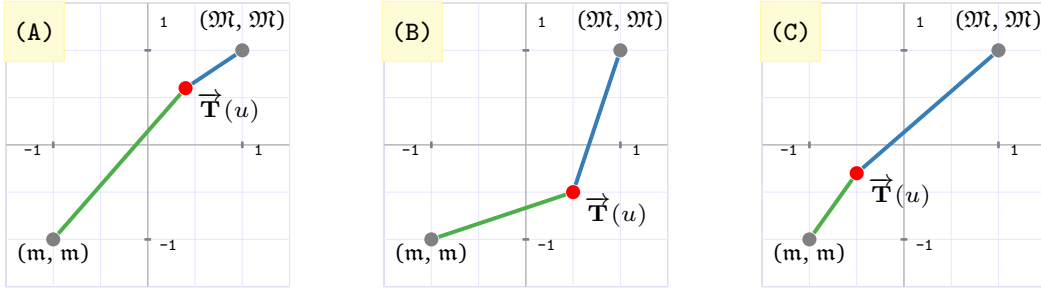


Figure 1. Illustration of the gullibility and the paranoia of a node u for which $|\vec{\Gamma}(u)| = n = 2$. In the case (A), we can see that $\vec{\mathbf{T}}(u)$ is closer to $\vec{\mathfrak{M}}_2 = (1, 1)$ than it is to $\vec{\mathfrak{m}}_2 = (-1, -1)$, we thus say that u is more gullible than paranoid. The inverse can be seen in the (C) case. The special case (B) illustrates a situation where $\vec{\mathbf{T}}(u)$ is equidistant from both vectors $\vec{\mathfrak{m}}_2$ and $\vec{\mathfrak{M}}_2$.

Definition 3.2. We say that a node v is *absolutely competent* if it is extremely trusted by all its trustors. i.e., $\forall u \in \overleftarrow{\Gamma}(v), t(u, v) = \mathfrak{M}$. And *absolutely incompetent* if it is extremely distrusted by all its trustors. i.e., $\forall u \in \overleftarrow{\Gamma}(v), t(u, v) = \mathfrak{m}$.

Using the same idea of euclidean distances to extreme values that we used to define the gullibility $g^+(u)$ and the paranoia $g^-(u)$ of a node u , we define the competence and the incompetence of a node v as follows:

3.3.3. Competence

Let $v \in \mathcal{N}$ be a node, and let $\overleftarrow{\mathbf{T}}(v) = (t(u_1, v), t(u_2, v), \dots, t(u_k, v))$ be a vector in I^k made of the trust values $t(u_i, v)$ that its trustors $u_i \in \overleftarrow{\Gamma}(v), i = 1, \dots, k$ put in it. We describe the *competence* of the node v using the function $c^+ : \mathcal{N} \mapsto [0, +1]$, that is defined as follows:

$$c^+(v) = \left[\overleftarrow{\Gamma}(v) \neq \emptyset \right] \left(1 - \frac{d(\overleftarrow{\mathbf{T}}(v), \overrightarrow{\mathfrak{M}}_k)}{\mathcal{R}\sqrt{k}} \right), \quad \text{where } k = |\overleftarrow{\Gamma}(v)|. \quad (3)$$

3.3.4. Incompetence

Taking this time the vector $\overrightarrow{\mathfrak{m}}_k$ as a reference, we describe the *incompetence* of a node v , using the function $c^- : \mathcal{N} \mapsto [0, +1]$, that is defined as follows:

$$c^-(v) = \left[\overleftarrow{\Gamma}(v) \neq \emptyset \right] \left(1 - \frac{d(\overleftarrow{\mathbf{T}}(v), \overrightarrow{\mathfrak{m}}_k)}{\mathcal{R}\sqrt{k}} \right), \quad \text{where } k = |\overleftarrow{\Gamma}(v)|. \quad (4)$$

Definition 3.3. A node u reciprocates trust from another node v , if $t(u, v) = t(v, u)$. The *propensity-to-reciprocate* trust (or *reciprocity*, for short) of the node u may be expressed by the distance between the vector $\overrightarrow{\mathbf{T}}(u) = (t(u, v_1), t(u, v_2), \dots, t(u, v_n))$ (outgoing trust from u to its trustees) and the vector $\overleftarrow{\mathbf{R}}(u) = (t(v_1, u), t(v_2, u), \dots, t(v_n, u))$ (incoming trust to u from its trustees). The *reciprocity* of the node u is described using the function $\tau : \mathcal{N} \mapsto [0, +1]$ that is defined as follows:

$$\tau(u) = \left[\overrightarrow{\Gamma}(u) \neq \emptyset \right] \left(1 - \frac{d(\overrightarrow{\mathbf{T}}(u), \overleftarrow{\mathbf{R}}(u))}{\mathcal{R}\sqrt{n}} \right), \quad \text{where } n = |\overrightarrow{\Gamma}(u)|. \quad (5)$$

3.4. Predicting trust using these metrics

Let u, v be two nodes in \mathcal{N} . We consider that the value $t(u, v)$ is influenced by these social traits that we have defined. This idea is supported by the following observations:

1. The more gullible u is, the more likely that $t(u, v)$ will be equal to \mathfrak{M} .
2. The more paranoid u is, the more likely that $t(u, v)$ will be equal to \mathfrak{m} .
3. The more competent v is, the more likely that $t(u, v)$ will be equal to \mathfrak{M} .
4. The more incompetent v is, the more likely that $t(u, v)$ will be equal to \mathfrak{m} .
5. The more u reciprocates trusts, the more likely that $t(u, v)$ will be equal to $t(v, u)$.

That $t(u, v)$ value in observations 1–5 above seems, so to speak, torn between the traits of both u and v . An analogy that comes to mind is that of a *tug of war* game; where people pull on opposite ends of a rope to bring the handkerchief in the middle to their side. This analogy leads us to our main hypothesis: the prevailing trait, be it of

the trustor or the trustee, would *pull* the trust value $t(u, v)$ toward its associated extreme value (See illustration on Fig. 2.). We, therefore, formulate the predicted trust value $t(u, v)$ as a mean of these extreme values (m , \mathfrak{M} , and $t(v, u)$) weighted by the values of the involved traits. That is, having:

$$\begin{cases} f^+(u, v) = \mathfrak{g}^+(u) + \mathfrak{c}^+(v), \\ f^-(u, v) = \mathfrak{g}^-(u) + \mathfrak{c}^-(v), \\ f^=(u, v) = (\mathfrak{r}(u) + \mathfrak{r}(v)) [t(v, u) \neq 0], \end{cases} \quad (6)$$

we estimate $t(u, v)$ to be equal to:

$$t(u, v) \approx \frac{\mathfrak{M} f^+(u, v) + m f^-(u, v) + t(v, u) f^=(u, v)}{f^+(u, v) + f^-(u, v) + f^=(u, v)}. \quad (7)$$

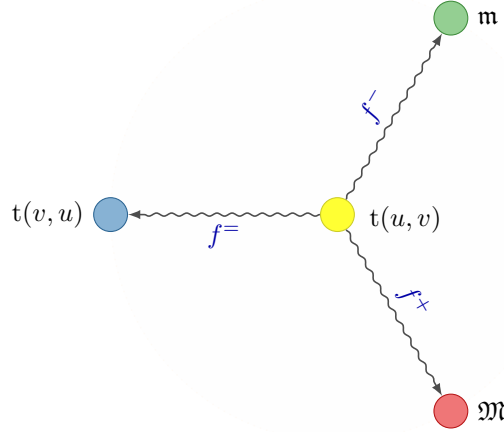


Figure 2. Predicting trust with a three-way tug of war analogy. The trust value $t(u, v)$ is affected by different *forces* that result from u 's and v 's traits. For instance, the gullibility of u and the competence of v , make $t(u, v)$ converge toward the maximal value \mathfrak{M} . The paranoia of u and the incompetence of v make it converge toward m . And finally, the reciprocity of both u and v make $t(u, v)$ converge toward $t(v, u)$ if this value exists.

The time complexity of this algorithm, when trying to predict the trust that a node u would put in another one v , is $\mathcal{O}(|\vec{\Gamma}(u)| + |\overleftarrow{\Gamma}(v)|)$.

3.5. Illustrative example

Consider the social graph illustrated in Figure 3. In this example, we take into consideration the direct neighbors of two nodes u and v , and try to predict how much would u trust (or distrust) v in a trust interval of $[-1, +1]$.

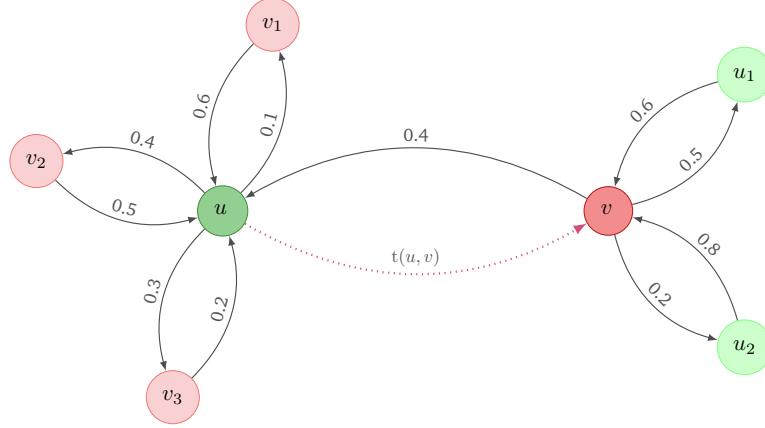


Figure 3. Example of trust inference using the gullibility (and paranoia) of u , the competence (and incompetence) of v , and their reciprocity.

In this example, we see that $\vec{\Gamma}(u) = \{v_1, v_2, v_3\}$ (i.e., $|\vec{\Gamma}(u)| = 3$). Therefore, to compute $g^+(u)$ and $g^-(u)$ we use $\vec{\mathfrak{M}}_3 = (1, 1, 1)$ and $\vec{\mathfrak{m}}_3 = (-1, -1, -1)$ respectively. Using Equations (1), (2), and (5) we find that: the gullibility of u is $g^+(u) = 0.628$, its paranoia is $g^-(u) = 0.364$, and its reciprocity is $\tau(u) = 0.836$.

As for the node v , we see that $\vec{\Gamma}(v) = \{u_1, u_2\}$ (i.e., $|\vec{\Gamma}(v)| = 2$). Thus, we use the vectors $\vec{\mathfrak{M}}_2 = (1, 1)$ and $\vec{\mathfrak{m}}_2 = (-1, -1)$. Applying the formulas in Equations (3), (4), and (5), we find that the competence of v is $c^+(v) = 0.842$, its incompetence is $c^-(v) = 0.149$, and its reciprocity is $\tau(v) = 0.790$.

Finally, in order to infer the trust that u would put in v , we refer to Equation (7) which gives us: $t(u, v) \approx 0.445$.

In the next section, we put this approach to trial, and see if the proposed hypothesis stands with real-world datasets.

4. Experimental evaluation

4.1. Datasets description

In order to validate the pertinence of the proposed approach, we have used the following real-world datasets during our experiments. The first three datasets are from the *Stanford Large Network Dataset Collection*², and the last one is from *Trustlet*³. Statistics about these datasets are given in Table 2.

²<http://snap.stanford.edu/data/>

³<http://www.trustlet.org/datasets/>

Bitcoin Alpha and OTC : Bitcoin is an anonymous cryptocurrency used by people and organizations to sell and buy goods and services. However, anonymity is often associated with the risk of fraud. These risks instigated the creation of some websites where users rate each other based on the trust that they put in them. We use two datasets from two sites: Bitcoin-Alpha and Bitcoin-OTC that were collected and scaled by Kumar et al. (2016) to fit trust (and distrust) in the interval $[-1, +1]$.

Wikipedia-Rfa : *Requests for adminship* (Rfa) are the formal procedure for Wikipedia members to be elected as administrators. A member (or someone on their behalf) submits a *Request for adminship* (Rfa) and members of the community express their opinions about the Rfa with a rating (+1 positive, 0 neutral, or -1 negative) and a comment. For their work, Kumar et al. (2016) analyzed these comments using the VADER sentiment engine (Gilbert, 2014) and generated a weighted and signed trust graph with weights in $[-1, +1]$.

Robots.net : Robots.net is a community of robotics enthusiasts who rank each other using the levels *observer*, *apprentice*, *journeyer*, or *master*. We mapped these levels to real numbers (0.1, 0.4, 0.7, and 0.9 respectively). The generated dataset, while unsigned, allowed us to see how the studied algorithms perform on a trust-only dataset.

Network	Nodes	Arcs	Reciprocity Ratio
Bitcoin-Alpha	3783	24186	59.57%
Bitcoin-OTC	5881	35592	56.89%
Wikipedia-Rfa	9654	104554	0.07%
Robots.net	1725	3596	7.78%

Table 2. Statistics about the used datasets. The *reciprocity ratio* column indicates how much trust is reciprocated in the network. That is, the percentage of arcs such that $t(u, v) = t(v, u)$.

4.2. Evaluated algorithms

To evaluate the merits of the proposed approach, we have compared the performances, the robustness, and the efficiency of the following algorithms that were also designed for weighted and signed networks.

Reciprocal (REC) is obviously the simplest. This algorithm is based on the assumption that if a node v trusts another node u , then u will probably trust v back as much as v trusts it. i.e., $t(u, v) = t(v, u)$ if the arc from v back to u exists, and 0 otherwise.

Bias and Deserve (BaD) we took $\text{DESERVE}(v)$ as the predicted trust value as described in Mishra and Bhattacharya (2011).

Fairness-Goodness (FxG) the trust from u to v is the product of the FAIRNESS of u by the GOODNESS of v as proposed by Kumar et al. (2016).

STAR we took the inferred trust value as proposed by Gao et al. (2016).

Trust by Agreement (AGR) we took the predicted trust value by agreement as described in Akilal et al. (2019).

Gullibility-Competence-Reciprocity (GCR) our own approach described in Section 3.

4.3. Performance evaluation metrics

Given a network $G(\mathcal{N}, \mathcal{E}, t)$ with $|\mathcal{N}| = N$, the performances of the above algorithms, were evaluated using the following metrics.

Mean Absolute Error (MAE) is the mean of the absolute differences between the ground truth values x_i and the inferred ones y_i , $i \in \{1, \dots, N\}$:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |x_i - y_i|.$$

Root Mean Squared Error (RMSE) is the root mean of the squared differences between the ground truth values x_i and the inferred ones y_i , $i \in \{1, \dots, N\}$:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2}.$$

Pearson Correlation Coefficient (PCC) which ranges between -1 and $+1$, indicates how the ground truth values x_i correlate with the predicted ones y_i . The more the PCC converges toward $+1$, the more the two values are correlated:

$$\text{PCC} = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}},$$

where \bar{x} (resp. \bar{y}) is the arithmetic mean of the values x_i (resp. y_i), $i \in \{1, \dots, N\}$.

4.4. Experiments

Two scenarios, largely inspired by the work done by Kumar et al. (2016), were studied during these experiments. In the first scenario, we are interested in predicting a yet-to-exist arc's weight knowing all other trust relations in the network. In the second, we remove a random batch of arcs and try to predict them all at once. That is, we try to predict trust relations with only some, or just a few, known relations from the network.

4.4.1. Leave-one-out predictions

This classic trust prediction task consists in removing an arc from the trust graph, and predicting its weight. It answers the question “How much would a node u (dis)trust another node v ” knowing all other trust relations in the network.

The five algorithms were run on every arc of the used datasets. For every dataset, we remove one arc at a time, predict its weight, put it back in the graph, and repeat the experiment with another arc. To measure the performances of these algorithms, we have calculated the MAE, RMSE, and PCC metrics for every pair of algorithm and dataset.

The results of these experiments are reported in Table 3. They show that the proposed approach outperforms the REC, FxG, BaD, and STAR algorithms on every metric (MAE, RMSE, PCC), and on every dataset, and is very close to the AGR approach.

	Bitcoin-Alpha	Bitcoin-OTC	Wikipedia-Rfa	Robots.net
REC	(0.12 , 0.27, 0.47)	(0.15, 0.32, 0.46)	(0.56, 0.63, 0.071)	(0.50, 0.60, 0.01)
FxG	(0.19, 0.33; 0.24)	(0.22, 0.38, 0.31)	(0.18, 0.24, 0.43)	(0.28, 0.33, -0.04)
BaD	(0.20, 0.34, 0.24)	(0.23, 0.40, 0.32)	(0.18, 0.23, 0.44)	(0.21, 0.31, 0.15)
STAR	(0.21, 0.32, 0.25)	(0.23, 0.35, 0.22)	(0.24, 0.32, 0.22)	(0.43, 0.50, 0.44)
AGR	(0.14, 0.24, 0.56)	(0.14 , 0.26 , 0.69)	(0.17 , 0.22 , 0.53)	(0.14 , 0.21 , 0.55)
GCR	(0.12 , 0.23 , 0.61)	(0.14 , 0.27, 0.67)	(0.17 , 0.22 , 0.53)	(0.16, 0.22, 0.46)

Table 3. Results from the leave-one-out tests. Inside each cell of this table is a tuple (MAE, RMSE, PCC) of the output of an algorithm (row) on a dataset (column). Lower MAE and RMSE, and higher PCC, are better. The best results being typeset in bold.

Efficiency comparison. To study how *fast* are the studied algorithms, we have measured the time that they take to perform a *leave-one-out* experiment on every arc of the used datasets. All these tests were run on an intel[®] i5-2450M CPU with 8GB of RAM. As shown in Figure 4, the proposed approach is almost as fast as the reciprocal algorithm which, obviously, is the fastest since its time complexity on a single arc is literally $\mathcal{O}(1)$. In fact, apart from the reciprocal algorithm, our approach is up to two orders of magnitude faster than the other four algorithms. Specifically, while the AGR and GCR approaches are close in terms of accuracy, the present GCR approach is by far more efficient than AGR.

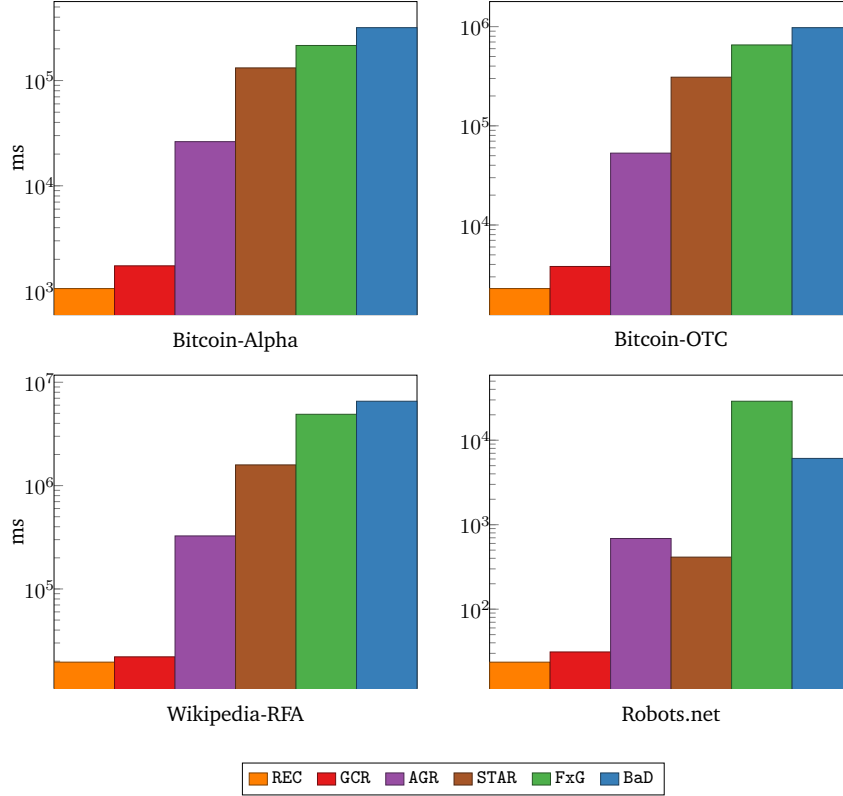


Figure 4. The time (in milliseconds) taken by each algorithm to perform a *leave-one-out* prediction on every arc of the used datasets. Note that the y-axis is logarithmic (Durations are expressed in powers of 10 milliseconds.)

4.4.2. Leave- $\mathcal{N}\%$ -out predictions

The purpose of this scenario, as explained by Kumar et al. (2016), is to study the robustness of prediction algorithms to network sparsity. By removing a random batch of arcs, prediction algorithms are expected to perform poorly since the trust relations that they would use for inferring others are no longer available.

For our experiments, we have randomly removed 10%, 20%, and so on, up to 90% arcs of every dataset, and have tried to predict their weights. For the sake of *fairness*, we have repeated these tests 100 times for every percent⁴, dataset, and algorithm. We then calculated the average MAE, RMSE, and PCC for each algorithm, with $\mathcal{N}\%$ removed arcs from every dataset. The results reported in Figure 5 show that, compared to the four other ones, the proposed GCR and AGR algorithms provide the best MAE, RMSE, and PCC in every case—in fact, their respective lines are even overlapping. These performances are, moreover, barely affected as we remove more arcs from the networks. In other words, our approach is reasonably robust to networks sparsity.

⁴Note that the $\mathcal{N}\%$ arcs removed in iteration $i + 1$ are different from those removed in iteration i . In every iteration, the arcs are *randomly* selected then removed from the graph.

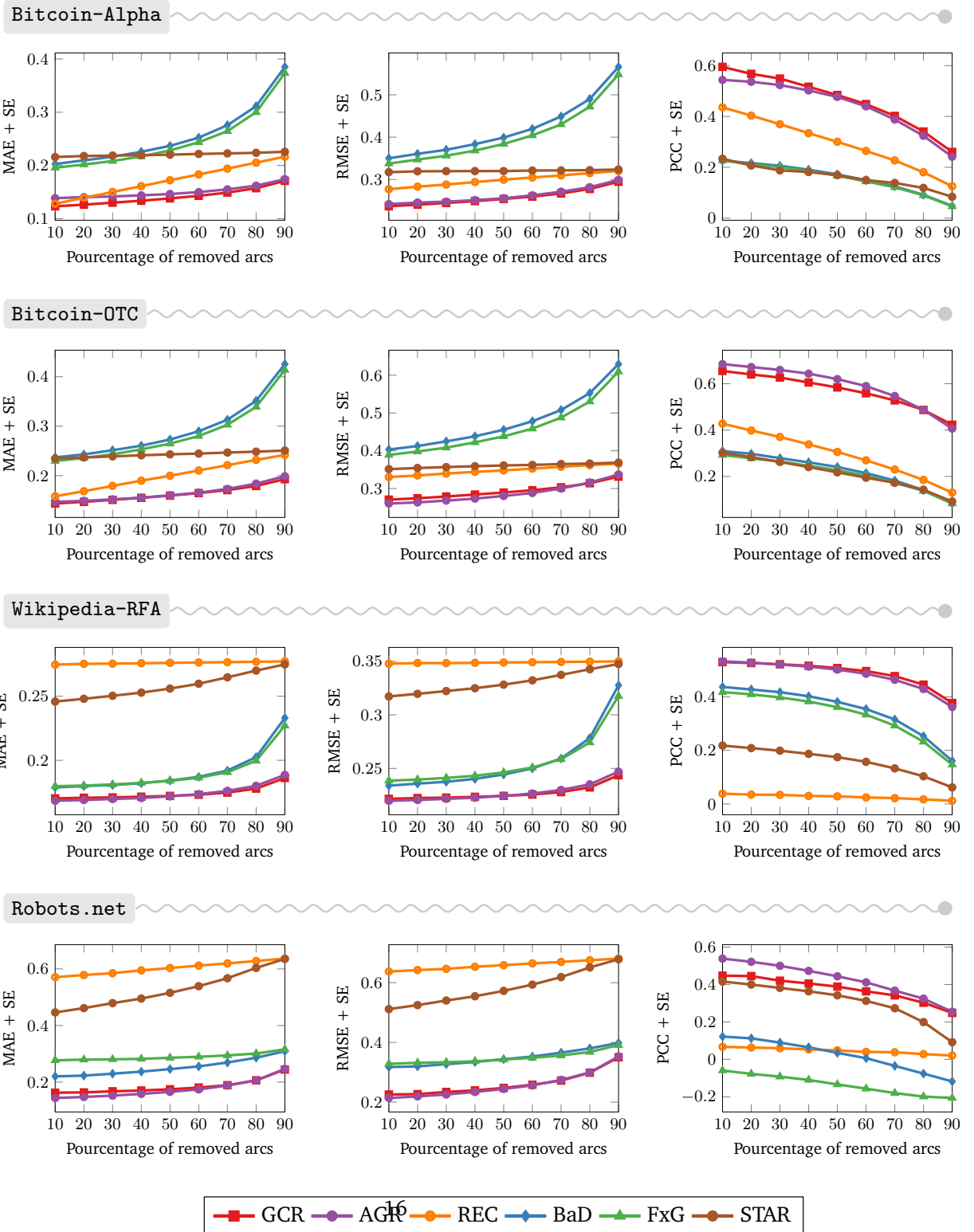


Figure 5. Leave- $N\%$ -out results. A grid of plots where: the x-axis are the percentage of removed arcs (going from 10% up to 90% in steps of 10%) from the datasets (rows). The three columns of plots show how the average MAE, RMSE, and PCC change as we remove more arcs from the datasets. Error bars (SEM) are too small to be seen.

5. Discussion

Back in the introduction, we have enumerated three desirable qualities in a trust prediction algorithm; namely, 1) accuracy, 2) robustness to network sparsity, and 3) speed. In this section, we discuss how these properties hold for the studied algorithms.

5.1. Accuracy

As shown in the previous section, the results from both the *leave-one-out* and the *leave-N%-out* experiments show that, on the four used datasets, the proposed approach and the AGR one are the most accurate among the studied algorithms. The difference in accuracy between the AGR and the GCR approaches is quite negligible as shown in Table 3 and the plots in Figure 5. The GCR approach outperforms the AGR one on the Bitcoin-Alpha dataset, and is *ex-æquo* with it on the Wikipedia-RFA one. On the other hand, the AGR approach presents a gain of 0.02 in MAE and one of 0.01 in RMSE on the Bitcoin-OTC, and Robots.net datasets. These differences should be negligible in a trust range equal to 2 (the adopted trust interval being $[-1, +1]$).

A precision worth making is that the nature of the networks population and activities are inherent factors that define trust relations emergence and evolution. For instance, the performances of the proposed approach are close to those of the reciprocal algorithm (REC) on the Bitcoin datasets (and only on these two datasets). That is because the Bitcoin datasets, contrary to the other two datasets, present a high reciprocity ratio (See Table 2). A property that allowed the reciprocal algorithm to perform quite well on the Bitcoin datasets, and poorly on the other datasets. A possible explanation for this might be that the *nature* of trust relations are different in these networks. Indeed, we can argue that since the Bitcoin datasets are about trust in trade, one would expect the buyer and the seller, once satisfied, to rate each-other equally —out of courtesy, if for nothing else. However, networks like Wikipedia and Robots.net are about knowledge and authority. In these networks, a very knowledgeable node does not feel the need, nor is it supposed, to return the favor by rating back its trustors more than they actually deserve. The proposed algorithm acknowledges *reciprocity* as a fact in social interactions, but it also takes into account two other aspects: *gullibility* and *competence*. The association —or the competition— between these three traits is what makes it more accurate than the other approaches.

5.2. Robustness

The slope of the curves in Figure 5 show that the performances of the GCR and AGR approaches are satisfyingly stable. Indeed, the MAE and RMSE values for these approaches seem unaffected by the number of removed arcs. Moreover, compared to the results of the other four algorithms, these values are better (smaller) in all cases.

A possible explanation for this might be found in the design of the algorithms themselves. Put simply, the more an approach depends on more arcs, the more it is likely to be affected by the unavailability of said arcs. Take, for example, the REC algorithm. This one depends only on one arc (the reciprocal one), the plots show, indeed, a somewhat stable behavior for this algorithm, yet this stability also holds when the algorithm performs poorly. Second, the global-metrics oriented algorithms (BaD, FxG) are the ones that are more affected by network sparsity as shown in Figure 5. The reason for this is that to compute a node's global metrics (BIAS/DESERVE, and FAIRNESS/GOODNESS), these

algorithms spread their computations to large areas of the networks. Therefore, as more arcs become unavailable, the reliability of these global metrics decreases, and so do the performances of these algorithms.

Interestingly, the STAR algorithm shows somewhat stable performances. At first, we expected that removing more arcs would *break* more trust paths, and since the STAR algorithm is propagative, its performances would decrease. However, the STAR algorithm actually sorts paths by their certainty. Thus, whenever a path is broken, another one is picked. With this said, a closer look at the plots in Figure 5 reveals that the performances of the GCR and AGR are, not only better than those of STAR, but are actually even better at 90% removed arcs than what STAR provides at 10% removed arcs. In summary, The GCR approach (and the AGR one) that we propose are more robust thanks to their main design choice: making prediction rely on the sole direct neighbors of the trustor, and those of the trustee.

5.3. Speed

Another desirable feature of trust prediction algorithms is speed. Without a doubt, the REC algorithm wins this race hands down. However, accuracy is more important than speed —A slightly delayed, yet wise, decision is *always* better than a hastened and bad one. Still, as shown in Figure 4, the proposed GCR approach is actually almost as fast as the reciprocal one, yet provides more accurate predictions than the other algorithms.

Again, the design of these algorithms is to thank (or to blame) for efficiency as well. The global-metrics oriented (BaD and FxG), and the STAR algorithms spread their calculations to larger areas of the networks and are hence slower. Whereas, the REC and the proposed GCR approaches are *localized* and thus faster. As for the AGR approach, although its performances and robustness are close to that of the proposed GCR approach, its efficiency seems poor in comparison. This lack of efficiency of the AGR algorithm is due to its heavy processing. Indeed, the time complexity of the AGR approach for computing a trust value from a node u to another one v is $\mathcal{O}(2|\vec{\Gamma}(u)||\overleftarrow{\Gamma}(v)|)$, whereas the time complexity of the GCR one, for the same task, is only of $\mathcal{O}(|\vec{\Gamma}(u)| + |\overleftarrow{\Gamma}(v)|)$.

Another point worth highlighting is that social networks are humming with activity and changes in trust relations. These changes should they happen —and they do— must be taken into account by trust prediction algorithms. Global-metrics-oriented and propagative approaches would require a lot of processing for every change, whereas localized approaches such as the REC, AGR, and GCR would only need to operate on the direct neighbors of the trust relation individuals.

As discussed in the introduction, speed *is* important. Yet, one might wonder how much accuracy can we sacrifice for speed. Indeed, the AGR algorithm presents a gain of about 0.02 in MAE, and about 0.01 in RMSE in two datasets (among four). We believe that these negligible gains pale in comparison to the big speed gains offered by the GCR approach (cf. Fig 4).

Storing and reusing the results of intermediate and heavy computations would probably be the best thing to do in a static trust graph with few to no activity or changes in topology and social relations. However, in the context of highly dynamic platforms such as social networks, where trust fluctuates, and where change is *the* constant rule, we believe that providing the most accurate, robustest, and quickest recommendations would require a real-time computation of trust using the freshest information available.

5.4. Closing remarks

We conclude this discussion with a few notes about some characteristics and limitations of the proposed GCR approach.

5.4.1. Trust properties

The novelty of the GCR approach makes us wonder whether the commonly known properties of trust hold for the proposed approach. Regarding the *subjectivity* property, the adoption of *personal* traits of both individuals is quite an admission (and use) of this aspect of trust. The approach stands on the idea that people are different (the “values” of their traits are) and that these traits affect trust. Furthermore, our work agrees with the *directed* aspect of trust since it considers trust graphs as wighted and directed ones, and does not take reciprocity for granted. Finally, the proposed approach is not *transitivity*-based. This aspect of trust as described by Guha et al. (2004) and others is, therefore, not affected.

5.4.2. Traits choice

In our work, we have considered only three traits among the multitude of traits that affect trust. Our choice was in fact driven by the following reasons:

1. These traits (gullibility, competence, and reciprocity) are simple to comprehend and to formulate. They are commonly known and often easily noticeable.
2. The extreme values associated with these traits are known without any computation. Indeed, the values \mathfrak{M} and \mathfrak{m} are given, and the reciprocate trust value $t(v, u)$, when known, requires no computation.
3. The effects of these traits on trust are known, as expressed in the 5 natural observations enunciated in Subsection 3.4 and that led to the proposed algorithm.

Otherwise, we profoundly think that there are many other traits that affect trust. As a matter of fact, we believe that if we know *all* these traits and their effects, then we should be able to predict trust with a very high accuracy. Therefore, we think that exploring more traits and including them in a future work would enhance even more the accuracy and the robustness of the proposed approach.

5.4.3. Traits dependence

Another aspect worth discussing about this approach is *traits dependence*. We should acknowledge that, for simplicity sake, the approach considered that the three traits were independent. Are they? First, the competence (or incompetence) of a trustee is indeed independent of the gullibility (or paranoia) of a trustor. A trait of a person barely affects a trait of another one from which he/she is *socially distant*. Someone being competent would not make another one (which he/she does *not* know) more gullible than he/she is, and vice versa.

However, the reciprocity of a trustor and his/her gullibility (or paranoia) are not proven to be independent. Even though the experimental results were satisfying, we still think that the accuracy of the algorithm could be improved by analyzing the relationship between these two traits in the same individual. In fact, this rises an interesting question: “Are gullible people more prone to reciprocate trust? How do gullibility and reciprocity correlate?” Statistical, and social, studies may shed more light on the subject, and their results would definitely help improve the proposed approach.

6. Conclusion

With the advent of social networks, predicting the amount of trust (or distrust) that a user should put in another one becomes a pressing need. Indeed, knowing how much should one trust or distrust someone else is essential for a better online presence. Unfortunately, in addition to their inability to properly handle distrust, most traditional prediction methods often suffer from some known limitations such as their inability to properly handle and predict distrust, their poor efficiency and robustness to network sparsity.

This paper presents a novel approach that aims to address these issues. Indeed, experimental evaluation on four real-world datasets have proven that the proposed algorithm is very efficient and satisfyingly accurate and robust to network sparsity. Its main strength is its astounding simplicity and intuitivity. Indeed, rather than opting for classical trust propagation, our algorithm involves some social traits in a kind of a *tug of war* game where the most prevailing social trait would have more influence on trust relations. This novel strategy does not inherit the limitations of most propagative approaches, as it is not subject to trust decay, opinion conflict, path dependence, and time complexity.

Furthermore, the proposed approach is easily extensible by taking into account more social traits that affect trust and involving them in the tug-of-war game that we have adopted. For instance, the *popularity* of a node, the *partiality* of a trustor, the *bandwagon* effect, to cite a few, are some social aspects worth investigating in a future work.

References

- Akilal, K., Slimani, H., Omar, M., 2019. A robust trust inference algorithm in weighted signed social networks based on collaborative filtering and agreement as a similarity metric. *Journal of Network and Computer Applications* 126, 123–132. doi:10.1016/j.jnca.2018.11.008.
- Beigi, G., Tang, J., Wang, S., Liu, H., 2016. Exploiting emotional information for trust/distrust prediction, in: *Proceedings of the 2016 SIAM International Conference on Data Mining, SIAM*. pp. 81–89. doi:10.1137/1.9781611974348.10.
- Braga, D.D.S., Niemann, M., Hellingrath, B., Neto, F.B.D.L., 2018. Survey on computational trust and reputation models. *ACM Computing Surveys* 51, 1–40. doi:10.1145/3236008.
- Chiang, K.Y., Hsieh, C.J., Natarajan, N., Dhillon, I.S., Tewari, A., 2014. Prediction and clustering in signed networks: a local to global perspective. *The Journal of Machine Learning Research* 15, 1177–1213.
- Chiluka, N., Andrade, N., Gkorou, D., Pouwelse, J., 2012. Personalizing eigentrust in the face of communities and centrality attack, in: *2012 IEEE 26th International Conference on Advanced Information Networking and Applications*, pp. 503–510. doi:10.1109/AINA.2012.48.
- DuBois, T., Golbeck, J., Srinivasan, A., 2011. Predicting trust and distrust in social networks, in: *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom)*, 2011 IEEE Third International Conference on, IEEE. pp. 418–424. doi:10.1109/PASSAT/SocialCom.2011.56.
- Gao, P., Miao, H., Baras, J.S., Golbeck, J., 2016. Star: semiring trust inference for trust-aware social recommenders, in: *Proceedings of the 10th ACM Conference on Recommender Systems, ACM*. pp. 301–308. doi:10.1145/2959100.2959148.
- Ghavipour, M., Meybodi, M.R., 2018. Trust propagation algorithm based on learning automata for inferring local trust in online social networks. *Knowledge-Based Systems* 143, 307–316. doi:10.1016/j.knsys.2017.06.034.
- Gilbert, C.H.E., 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text.
- Golbeck, J., 2005a. Personalizing applications through integration of inferred trust values in semantic web-based social networks, in: *Semantic Network Analysis Workshop at the 4th International Semantic Web Conference*, p. 30.
- Golbeck, J.A., 2005b. *Computing and Applying Trust in Web-based Social Networks*. Ph.D. thesis. College Park, MD, USA. AAI3178583.

- Graham, R.L., Knuth, D.E., Patashnik, O., 1994. Concrete mathematics a foundation for computer science. Addison-Wesley.
- Guha, R., Kumar, R., Raghavan, P., Tomkins, A., 2004. Propagation of trust and distrust, in: Proceedings of the 13th International Conference on World Wide Web, ACM, New York, NY, USA. pp. 403–412. doi:10.1145/988672.988727.
- Guo, S.L., Lumineau, F., Lewicki, R.J., et al., 2017. Revisiting the foundations of organizational distrust. *Foundations and Trends® in Management* 1, 1–88. doi:10.1561/3400000001.
- Hawley, K., 2013. Trust, distrust and commitment. *Noûs* 48, 1–20. doi:10.1111/nous.12000.
- Hu, Y., Wang, S., Ren, Y., Choo, K.K.R., 2018. User influence analysis for github developer social networks. *Expert Systems with Applications* 108, 108 – 118. doi:10.1016/j.eswa.2018.05.002.
- Huang, H., Dong, Y., Tang, J., Yang, H., Chawla, N.V., Fu, X., 2018. Will triadic closure strengthen ties in social networks? *ACM Trans. Knowl. Discov. Data* 12, 30:1–30:25. doi:10.1145/3154399.
- Jiang, W., Wang, G., Bhuiyan, M.Z.A., Wu, J., 2016a. Understanding graph-based trust evaluation in online social networks: Methodologies and challenges. *ACM Computing Surveys* 49, 10:1–10:35. doi:10.1145/2906151.
- Jiang, W., Wang, G., Wu, J., 2014. Generating trusted graphs for trust evaluation in online social networks. *Future Generation Computer Systems* 31, 48 – 58. doi:10.1016/j.future.2012.06.010. special Section: Advances in Computer Supported Collaboration: Systems and Technologies.
- Jiang, W., Wu, J., Li, F., Wang, G., Zheng, H., 2016b. Trust evaluation in online social networks using generalized network flow. *IEEE Transactions on Computers* 65, 952–963. doi:10.1109/TC.2015.2435785.
- Jøsang, A., Pope, S., 2005. Semantic constraints for trust transitivity, in: Proceedings of the 2nd Asia-Pacific conference on Conceptual modelling-Volume 43, Australian Computer Society, Inc. pp. 59–68.
- Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H., 2003. The eigentrust algorithm for reputation management in p2p networks, in: Proceedings of the 12th International Conference on World Wide Web, ACM, New York, NY, USA. pp. 640–651. doi:10.1145/775152.775242.
- de Kerchove, C., Dooren, P.V., 2008. The PageTrust algorithm: How to rank web pages when negative links are allowed? pp. 346–352. doi:10.1137/1.9781611972788.31.
- Kim, Y.A., Ahmad, M.A., 2013. Trust, distrust and lack of confidence of users in online social media-sharing communities. *Knowledge-Based Systems* 37, 438–450. doi:10.1016/j.knosys.2012.09.002.
- Kleinberg, J.M., 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)* 46, 604–632. doi:10.1145/324133.324140.
- Knuth, D.E., 1992. Two notes on notation. *The American Mathematical Monthly* 99, 403–422. doi:10.2307/2325085.
- Kumar, S., Spezzano, F., Subrahmanian, V., Faloutsos, C., 2016. Edge weight prediction in weighted signed networks, in: Data Mining (ICDM), 2016 IEEE 16th International Conference on, IEEE. pp. 221–230. doi:10.1109/ICDM.2016.0033.
- Kunegis, J., Preusse, J., Schwagereit, F., 2013. What is the added value of negative links in online social networks?, in: Proceedings of the 22Nd International Conference on World Wide Web, ACM, New York, NY, USA. pp. 727–736. URL: <http://doi.acm.org/10.1145/2488388.2488452>, doi:10.1145/2488388.2488452.
- Kurdi, H.A., 2015. Honestpeer: An enhanced eigentrust algorithm for reputation management in p2p systems. *Journal of King Saud University - Computer and Information Sciences* 27, 315 – 322. doi:10.1016/j.jksuci.2014.10.002.
- Kuter, U., Golbeck, J., 2010. Using probabilistic confidence models for trust inference in web-based social networks. *ACM Transactions on Internet Technology* 10, 1–23. doi:10.1145/1754393.1754397.
- Lewicki, R.J., McAllister, D.J., Bies, R.J., 1998. Trust and distrust: New relationships and realities. *Academy of management Review* 23, 438–458.
- Li, T., Gupta, B.B., Metere, R., 2018. Socially-conforming cooperative computation in cloud networks. *Journal of Parallel and Distributed Computing* 117, 274–280. doi:10.1016/j.jpdc.2017.06.006.
- Liu, G., Wang, Y., Orgun, M.A., et al., 2011. Trust transitivity in complex social networks., in: AAI, pp. 1222–1229.
- Mao, C., Xu, C., He, Q., 2019. A cost-effective algorithm for inferring the trust between two individuals in social networks. *Knowledge-Based Systems* 164, 122–138. doi:10.1016/j.knosys.2018.10.027.
- Massa, P., Avesani, P., 2005. Controversial users demand local trust metrics: An experimental study on opinions.com community, in: Proceedings of the 20th National Conference on Artificial Intelligence - Volume 1, AAAI Press. pp. 121–126.
- Massa, P., Avesani, P., 2007. Trust metrics on controversial users: Balancing between tyranny of the majority. *International Journal on Semantic Web and Information Systems (IJSWIS)* 3, 39–64.
- Mishra, A., Bhattacharya, A., 2011. Finding the bias and prestige of nodes in networks based on trust scores,

- in: Proceedings of the 20th international conference on World wide web, ACM. pp. 567–576. doi:10.1145/1963405.1963485.
- Page, L., Brin, S., Motwani, R., Winograd, T., 1999. The PageRank citation ranking: Bringing order to the web. Technical Report. Stanford InfoLab.
- Papaoikonomou, T., Kardara, M., Tserpes, K., Varvarigou, T., 2013. The strength of negative opinions, in: Iliadis, L., Papadopoulos, H., Jayne, C. (Eds.), *Engineering Applications of Neural Networks*, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 90–99. doi:10.1007/978-3-642-41016-1_10.
- Pavleska, T., Blažič, B.J., 2016. User bias in online trust systems: aligning the system designers' intentions with the users' expectations. *Behaviour & Information Technology* 36, 404–421. doi:10.1080/0144929x.2016.1239761.
- Robbins, B.G., 2016. What is trust? a multidisciplinary review, critique, and synthesis. *Sociology Compass* 10, 972–986. doi:10.1111/soc4.12391.
- Rousseau, D.M., Sitkin, S.B., Burt, R.S., Camerer, C., 1998. Not so different after all: A cross-discipline view of trust. *Academy of Management Review* 23, 393–404. doi:10.5465/amr.1998.926617.
- Ruan, Y., Durrresi, A., 2016. A survey of trust management systems for online social communities – trust modeling, trust inference and attacks. *Knowledge-Based Systems* 106, 150 – 163. doi:10.1016/j.knosys.2016.05.042.
- Shahriari, M., Jalili, M., 2014. Ranking nodes in signed social networks. *Social Network Analysis and Mining* 4, 172. doi:10.1007/s13278-014-0172-x.
- Shneiderman, B., 2015. Building trusted social media communities: A research roadmap for promoting credible content, in: *Roles, trust, and reputation in social media knowledge markets*. Springer, pp. 35–43. doi:10.1007/978-3-319-05467-4_2.
- Tang, J., Chang, Y., Aggarwal, C., Liu, H., 2016. A survey of signed network mining in social media. *ACM Computing Surveys* 49, 42:1–42:37. doi:10.1145/2956185.
- Tang, J., Hu, X., Liu, H., 2014. Is distrust the negation of trust?: The value of distrust in social media, in: *Proceedings of the 25th ACM Conference on Hypertext and Social Media*, ACM, New York, NY, USA. pp. 148–157. doi:10.1145/2631775.2631793.
- Tang, J., Liu, H., 2015. Trust in social media. *Synthesis Lectures on Information Security, Privacy, & Trust* 10, 1–129. doi:10.2200/S00657ED1V01Y201507SPT013.
- Zhao, H., Xu, X., Song, Y., Lee, D.L., Chen, Z., Gao, H., 2018. Ranking users in social networks with higher-order structures.
- Ziegler, C.N., 2013. *Trust Propagation Models*. Springer International Publishing, Cham. pp. 99–131. doi:10.1007/978-3-319-00527-0_7.
- Ziegler, C.N., Golbeck, J., 2015. Models for trust inference in social networks, in: *Propagation Phenomena in Real World Networks*. Springer, pp. 53–89.
- Zolfaghar, K., Aghaie, A., 2010. Mining trust and distrust relationships in social web applications, in: *Intelligent Computer Communication and Processing (ICCP)*, 2010 IEEE International Conference on, IEEE. pp. 73–80. doi:10.1109/ICCP.2010.5606460.
- Zollo, F., Quattrocioni, W., 2018. Misinformation spreading on facebook, in: *Computational Social Sciences*. Springer International Publishing, pp. 177–196. doi:10.1007/978-3-319-77332-2_10.