A Robust Trust Inference Algorithm in Weighted Signed Social Networks based on Collaborative Filtering and Agreement as a Similarity Metric

Karim Akilal, Hachem Slimani, Mawloud Omar

A Robust Trust Inference Algorithm in Weighted Signed Social Networks based on Collaborative Filtering and Agreement as a Similarity Metric

Karim Akilal, Hachem Slimani, Mawloud Omar

Laboratoire d'Informatique Médicale, Faculté des Sciences Exactes, Université de Bejaia, 06000 Bejaia, Algerie

Abstract

Trust is a very significant notion in social life, and even more in online social networks where people from different cultures and backgrounds interact. Weighted Signed Networks (WSNs) are an elegant representation of social networks, since they are able to encode both positive and negative relations, thus allow to express trust and distrust as we know them in the real world. While many trust inference algorithms exist for traditional unsigned networks, distrust makes it hard to adapt them to WSNs. In this paper, we propose a new unsupervised trust inference algorithm based on collaborative filtering (CF), where we consider the trustors as users, the trustees as items, and use *agreement* as a local similarity metric to predict trust values in signed, and unsigned, networks. In addition to its prediction performances, experiments on four real-world datasets show that our algorithm is very robust to network sparsity.

Keywords: Online Social Network, Trust Inference, Distrust, Weighted Signed Network, Collaborative Filtering, Agreement.

1. Introduction

Trust, like probably anything that matters in life, has two conflicting aspects: on the one hand, it is fundamental to any social relationship (Robbins, 2016; Kleinberg, 1999), and on the other hand it implies taking risks, which may range from slight disappointments to big betrayals (Mayer et al., 1995; Jones, 1996). To mitigate these risks, we have two options: *a*) Avoid trusting altogether, therefore not taking any risk, but then again miss out on so many opportunities, *b*) Learn when, who, and how much to trust, and minimize the risks that it implies.

While trust may seem intuitive or natural in real life, it *is* complex and difficult to assess online. This difficultly may be attributed to several factors such as: the lack of the proximity and the clues that assist us in real life; the big differences in cultures and

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

motives; and the huge volumes of conflicting information to which we are exposed online. So if technology has the power to enable us to communicate and trust despite the distances, it should also have the power —or even the duty— to help us assess trust despite theses distances. And unsurprisingly, many research efforts have been undertaken in recent years in modeling, processing, and predicting trust (Cho et al., 2015; Sherchan et al., 2013; Tang et al., 2016; Ruan and Durresi, 2016; Jiang et al., 2016a).

Trust, however, is a double sided coin. Trying to predict trust alone, without considering its counterpart; *distrust*, is often not enough to assist users (DuBois et al., 2011). A good recommendation should help the user know whom to trust *and* whom to *not* trust. If they were about to share sensitive information online, a good recommendation would tell them whom they may share it with and; more importantly; whom they should not. If they get their news online, they should be able to filter the fake from the true.

Nevertheless, distrust presents some challenges that make adapting most of the work that has been done on unsigned networks (those that do not consider distrust) hard, or even impossible (Guha et al., 2004; Chiang et al., 2014; Tang et al., 2016). For example, while trust is somewhat transitive (Golbeck, 2005b), distrust is not. What, in fact, may be deduced when *Alice* distrusts *Bob* who distrusts *Carol*? Even though the old saying *"The enemy of my enemy is my friend"* is sometimes true, it is not always the case. Indeed, what if *Carol* is worse than *Bob*? *Alice* would not trust her. A fact supported by empirical evidence (Tang et al., 2014).

While some early trust modelizations and inference algorithms completely ignore distrust (Ziegler, 2013), the recent trend in research on social networks is to fully and rightfully *embrace* distrust. Indeed, many recent efforts integrate it to solve, and improve on already existing solutions to problems such as sign prediction, link prediction, edge weight prediction, node ranking and other tasks all pertaining to WSNs (Tang et al., 2016). This widespread interest in distrust as a *thing* clearly confirms that it *"is not the mere absence of trust"* as Hawley (2013) puts it.

Likewise, in our work, we do consider distrust as a crucial aspect in social life, and rather than relying on transitivity, which is problematic for distrust, we turn toward a reasoning that stands on the intuition that:

1. If two trustors agree about a set of common trustees, then they may also agree on a future common trustee.

2. If two trustees are making the consensus among their common trustors, then they may also be equally trusted by a future common trustor.

We expand on this intuition by emitting the following hypothesis. Given two nodes u and v in a WSN, to predict the trust that the trustor u would put in the trustee v, we can:

1. Average the trusts that v receives from its trustors weighted by their similarities with u.

2. Average the trusts that u puts in its trustees weighted by their similarities with v.

Such a formulation is in fact the basis of user–user and item–item collaborative filtering (CF) problems (Aggarwal et al., 2016), where, in our case, the trustors are the users, the trustees are the items, and the similarity is expressed in terms of agreements. The challenge of this kind of approaches is two-fold:

a) Find a metric by which we can accurately express the similarity between users, and between items.

b) Ensure that this metric will be able to handle very sparse networks where finding similar users, or similar items, is hard because there is not enough information to infer from.

Our contributions which aim to address these challenges are summarized as follows:

1. Introducing *agreement* as a similarity metric between trustors, and between trustees.
2. Proposing a formal definition of agreement that is able to handle sparse networks.
3. Proposing a trust inference algorithm based on agreement, that not only is able to predict trust and distrust values, but also proves to be robust to network sparsity.

The rest of this paper is structured as follows. In Section 2, we give a brief review of some related work. In Section 3, we present our approach starting from the idea of *agreement* to its use as the main ingredient in our trust inference algorithm. In Section 4, in order to validate our main hypothesis, we conduct a series of experiments to analyze the accuracy of our algorithm and its robustness to networks sparsity. After a discussion of the results in Section 5, we conclude this study in Section 6 with some perspectives of future work.

2. Related work

According to Tang and Liu (2015), trust can be described by *a*) global metrics which associate to every node of the network some characteristics such as how popular, trustworthy, or sociable it is, or *b*) by local metrics which describe relations between two nodes in the network (e.g., how does a node u trusts another one v). Predicting these trust metrics can be seen as *a*) a supervised task, where it is treated as a classification problem ; or *b*) an unsupervised task, which relies only on the information within the social graph to predict unknown trust values (or signs). Furthermore, these tasks may be supported by additional information such as interaction data among actors in the social network (Xiang et al., 2010; Jones et al., 2013; Huang et al., 2018), emotions (Beigi et al., 2016) or simply rely only on what the network already offers, i.e., the existing trust relations.

This quick overview of trust inference tasks gives us a rough idea of how wide is the subject, and the hierarchy becomes even deeper the more we add other classifications criteria. For the sake of brevity, and to accurately position our work, we refer the reader to some studies on the subject such as those by Jiang et al. (2016a) and Tang et al. (2016). We focus in what follows on some examples of graph-based unsupervised methods with no additional interactions data, and how the advent of signed networks adds more challenges to the question.

Among the unsupervised algorithms that operate on the sole network, without any additional communication information, we can cite some of the populars ones such as: TidalTrust (Golbeck, 2005a), MoleTrust (Massa and Avesani, 2007) and GFTrust (Jiang et al., 2016b). These local metrics algorithms are built on trust propagation, of which

Guha et al. (2004) stated the four atomic rules, namely: direct propagation (transitivity), transpose, co-citation, and trust coupling. As for global metrics, algorithms such as HITS (Kleinberg, 1999), PageRank (Page et al., 1999), or Eigentrust (Kamvar et al., 2003) are examples of methods that compute global attributes for the nodes.

The forecited works were designed to operate on unsigned networks and ignore, by design, the notion of distrust or consider it as the absence of trust. Recently, however, researchers started to consider distrust. For instance, Zolfaghar and Aghaie (2010); Mishra and Bhattacharya (2011) introduced new global metrics for signed networks, while Shahriari and Jalili (2014) adapted the HITS, and the PageRank algorithms to include negative links. Very recently, Kumar et al. (2016) defined two new global metrics: *fairness* and *goodness*, and a way to infer local trust simply by multiplying the fairness of the trustor by the goodness of trustee. Speaking of local metrics, Gao et al. (2016) proposed STAR, a propagation-based algorithm for trust and distrust prediction using a 2D semiring framework. It is worth noting that most propagation–based methods suffer from some limitations such as trust decay on long paths, and path dependences (Jiang et al., 2016b)

In addition to propagation-based approaches, some other works are built on various social theories such as homophily, structural balance, status, and others which Yap and Harrigan (2015) discuss and compare. With regard to homophily, Ziegler and Golbeck (2007) demonstrated that there is a strong correlation between trust and interest similarity. That is, if two individuals share the same interests and tastes, they are likely to trust each other consequently. Borzymek et al. (2009) expanded on this correlation to infer trust values between similar users; while Korovaiko and Thomo (2013) explored various similarity factors of users such as: their ratings to reviews, their shared interests, their common trustees, to predict trust in a supervised setting.

Structural balance theory (Heider, 1946; Cartwright and Harary, 1956) and Status theory (Leskovec et al., 2010) explore the relations in triads of undirected networks for the former, and directed ones for the latter. In a nutshell, structural balance stands on the assumptions that *"the friend of my friend is my friend"* and *"the enemy of my enemy is my friend"*. Status, on the other hand, is defined as a statement of the trustee's *status* by the trustor, such that if a positive link goes from u to v, then v has a higher status than u, and if the link is negative than u has a higher status than v (Tang et al., 2015). Both theories are empowering many recent efforts. For example, Wang et al. (2015) provided a mathematical model for the status theory, and an algorithm for trust prediction in a graph where a link from a node u to another v means that u trusts v, while Yuan et al. (2017) developed a method using both theories for link sign prediction. It is also worth noting that experiments conducted by Tang et al. (2015) on datasets from Epinion and Slashdot show that more than 90% of the triads in these networks are consistent with both theories. Such a finding reinforce the solidity of these theories which may, therefore, be used in sign and link predictions. However, to our knowledge, extended versions of these theories to weighted signed networks are yet to be formulated.

Finally, another category of methods, which we may describe as one at the crossroad of the previous two, includes works based on collaborative filtering (CF), which generally aim to predict how would a user rate an item by using already observed ratings. Collaborative filtering methods are generally divided into two main categories: model-based methods, which leverage machine learning techniques; and memory-based ones, in which the rating by a user u of an item p is predicted using information from their

neighborhood. This neighborhood, as described in (Aggarwal et al., 2016), is defined in two ways:

- User-based: the ratings provided by like-minded users of a target user u are used to make a recommendation for u. For example, to predict how would *Alice* rate the book *"The Iliad"*, we rank users who rated this book by their similarities to *Alice* and return an aggregated rating prediction.
- Item-based: the rating by the target user u on similar items to the target item p are used to predict how would the user u rate the item p. In this method, we look for books similar to *"The Iliad"*, which *Alice* already rated, and rank them based on their similarities to *"The Iliad"* and return an aggregated rating prediction.

To the best of our our knowledge, only a few works used memory-based CF methods in trust predicting. For instance, Garakani and Jalali (2014) proposed a CF algorithm for trust prediction using the NHSM similarity metric by Liu et al. (2014), while Ghodousi and Hamzeh (2015) proposed a CF-based approach using the Pearson Correlation Coefficient (PCC) as a similarity measure.

To be effective, CF methods have to tackle some issues. First, they need to define a similarity metric between the nodes of a network that is accurate enough to weight the recommendations of a neighbor. To this end, we introduce *agreement* as a similarity metric. Second, these methods have to be robust to network sparsity, which can be described as a situation where the target user rates only a few items, or the target item is rated by only a few users. For this purpose, we propose a two-way approach (user-user and item-item) and a *bootstrapping* factor to alleviate this problem. In summary, our algorithm can be described as a two-way CF-based one that uses agreement as a similarity metric. Details of this approach are given in the following section.

3. Our proposed approach

3.1. Notation and preliminaries

Two important facts about trust should be stated before choosing a model:

1. Trust may be positive or negative (distrust).
2. Trust is more than a binary concept. We say that *Alice* trusts *Bob* a lot, and slightly distrusts *Carol*. That is, each state has different continuous *shades*.

With this in mind, we represent a social network using a directed weighted graph, where each node represents an individual, and each arc going from a node u to another node v, indicates that u trusts (or distrusts) v. The weight of such an arc is the signed amount of trust that the source puts in the target. Table 1 summarizes the notation that we will adopt throughout this paper.

Notation	Meaning
$\mathcal{G}(\mathcal{N}, \mathcal{E}, \mathcal{W})$	A weighted directed graph $\mathcal{G}$ with nodes in $\mathcal{N}$ connected by arcs in $\mathcal{E}$ that are weighted using the mapping $\mathcal{W}$.
$\mathcal{G}^T$	Converse, or transpose, of the graph $\mathcal{G}$
$\mathcal{W}(p, q)$	Weight of the arc going form node p to node q.
$\overrightarrow{\Gamma}(p)$	Set of the trustees of the node p.
$\overleftarrow{\Gamma}(q)$	Set of the trustors of the node q.
R	Trust range. $R = \max(trust) - \min(trust)$.
$\overrightarrow{\mathtt{A}_{\mathcal{G}}}(u, v, w)$	Agreement of the trustors u and v about the trustee w.
$\overleftarrow{\mathtt{A}_{\mathcal{G}}}(w, z, u)$	Agreement on the trustees w and z by the trustor u.
$\overrightarrow{\mathtt{AGR}_{\mathcal{G}}}(u, v)$	Aggregated agreement score between the trustors u and v
$\overleftarrow{\mathtt{AGR}_{\mathcal{G}}}(u, v)$	Aggregated agreement score between the trustees u and v

Table 1. Notation used in this paper

3.2. Problem definition

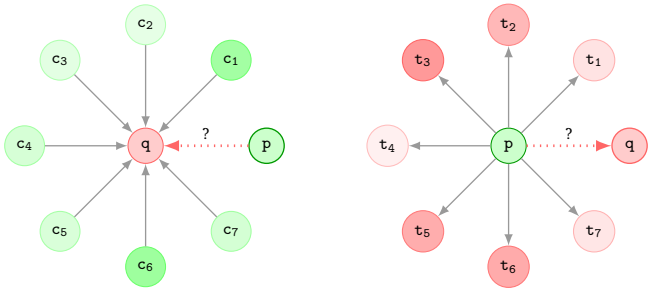
Given a social network represented by a directed and weighted graph $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathcal{W})$, where $\mathcal{N}$ is a set of users (or nodes), $\mathcal{E}$ a set of arcs between nodes of $\mathcal{N}$, and $\mathcal{W} : \mathcal{E} \mapsto [-1, +1]$ a mapping that associates to each arc (p, q) a weight w that describes how much p trusts ($w > 0$) or distrusts ($w < 0$) q. The problem that this paper tries to solve is to predict how much would a node p (dis)trust another node q, and that when all, or only some, of the remaining weights are known.

3.3. The basic idea behind our approach

As illustrated in Fig. 1, we summarize our idea as follows: given two nodes p and q, to predict how much p trusts, or distrusts, q, we ask ourselves these two questions:

Q1: How do other trustors of q trust it, and how much similar to p are they?

Q2: How does p trust its others trustees and how much similar to q are they?



(a) Trustors similarity (b) Trustees similarity

Figure 1. How should p trust q knowing how others trust q? And how should q be trusted by p knowing how p trusts others? A single question with two possible answers: the first coming from q's trustors, the second from p's trustees.

To predict the trust value $\mathcal{W}(p, q)$, our approach may be summarized in these steps:

1. Take the average of the trust that q receives from its trustors, which we weight by their respective similarities with p, and return, along this result, a confidence score.
2. Take the average of the trusts that p puts in its trustees weighted by their respective similarities with q, and return, along this result, a confidence score.
3. Average the results of the previous steps weighted by their respective confidence scores.

In an unsigned binary network, where trust is materialized by the existence of an arc from a node to another, and distrust by the absence of such an arc, a traditional metric such as the Jaccard coefficient may be used as a similarity metric.

As defined in Eq. (1), the Jaccard coefficient on trustees is the number of common trustees of two users divided by the total number of their unique trustees (Tang and Liu, 2015).

$$\overrightarrow{J}(p, p') = \frac{\left|\overrightarrow{\Gamma}(p) \cap \overrightarrow{\Gamma}(p')\right|}{\left|\overrightarrow{\Gamma}(p) \cup \overrightarrow{\Gamma}(p')\right|}. \tag{1}$$

However, in the case of WSNs, we can not rely on the *number* of common trustees (or trustors) as a similarity indication between trustors (or trustees respectively). For instance, let us consider the scenario depicted in Fig 2.

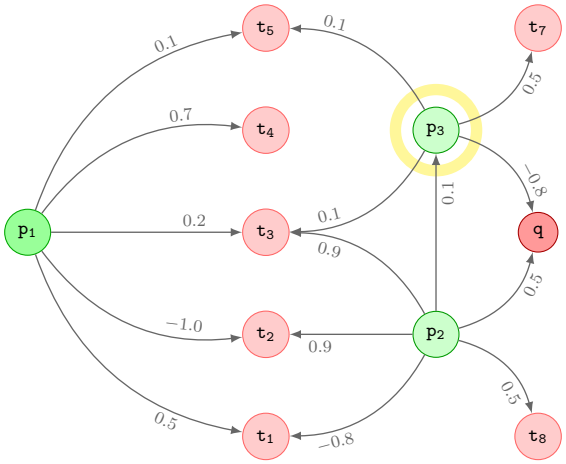


Figure 2. How much would $\mathtt{p}_1$ trust $\mathtt{q}$, knowing how much p_2 and p_3 trust $\mathtt{q}$ and *agree* more or less with $\mathtt{p}$ about some or all of his/her trustees? *"Quality over quantity"* should be the answer.

As we can see, p_1 shares 3 trustees with p_2, and 2 with p_3. However, were we to predict the trust that p_1 would put in q we would *intuitively* choose p_3 as a reference instead of p_2. The reason is obvious from the graph: p_1 tends to *agree* more with p_3. The arcs' weights are more important than their number.

Our hypothesis is, thus, formulated as follows: two trustors agreeing on a set of common trustees are more likely to agree on a future common trustee. Symmetrically, two trustees being agreed upon by a set of common trustors are more likely to be equally trusted by a future common trustor.

In what follows, we present a formal definition of agreement by two trustors, and about two trustees. Then, we propose an algorithm to test the validity of our main hypothesis.

3.4. Agreement as a similarity metric

First, we define *agreement* as a metric that measures how similarly two trustors trust a specific trustee, and how similarly two trustees are trusted by a specific trustor. Then, we expand on this definition to propose *aggregated agreement scores* between two trustors, and between two trustees.

3.4.1. Trustors and trustees agreement

Given a network $\mathcal{G}(\mathcal{N}, \mathcal{E}, \mathcal{W})$, and a constant $\lambda > 0$, the agreement of two trustors u and v about a trustee w, and which we denote as $\overrightarrow{\mathtt{A}_{\mathcal{G}}}(u, v, w)$, is expressed as follows:

$$\overrightarrow{\mathtt{A}_{\mathcal{G}}}(u, v, w) = exp\Big(\frac{-\lambda|\mathcal{W}(u, w) - \mathcal{W}(v, w)|}{R + \epsilon - |\mathcal{W}(u, w) - \mathcal{W}(v, w)|}\Big), \tag{2}$$

where $\epsilon \to 0^+$ and R is the trust range ($R = 2$ in our case, since we choose trust to be in $[-1, +1]$). As we can see in Eq. (2), the agreement decays exponentially as the absolute difference between the two trust values increases, and will reach 0 when the trusts values are at the opposite borders of the trust range.

We call the parameter λ an *agreement decay factor* since it decides on how fast the agreement between two trustors decays as the difference between their trusts in a common trustee gets bigger.

Similarly, we say that two trustees w and z are agreed upon by a common trustor u, if the latter trusts them somewhat equally. Thus, we define:

$$\overleftarrow{\mathtt{A}_{\mathcal{G}}}(w, z, u) = exp\Big(\frac{-\lambda|\mathcal{W}(u, w) - \mathcal{W}(u, z)|}{R + \epsilon - |\mathcal{W}(u, w) - \mathcal{W}(u, z)|}\Big). \tag{3}$$

Note that merging the two previous definitions into one is straightforward by noticing that:

$$\overleftarrow{\mathtt{A}_{\mathcal{G}}}(w, z, u) = \overrightarrow{\mathtt{A}_{\mathcal{G}^T}}(w, z, u), \tag{4}$$

where $\mathcal{G}^T$ is the converse, or the transpose, of the graph $\mathcal{G}$. i.e., the graph resulting from reversing the orientation of all the arcs of $\mathcal{G}$.

3.4.2. Aggregated trust agreement

Inspired by the Jaccard coefficient mentioned earlier, the aggregated agreement of two trustors u and v about all their trustees can, thus, be calculated as the sum of their agreements about their common trustees divided by the number of all their trustees. That is:

$$\overrightarrow{\mathtt{AGR}_{\mathcal{G}}}(u,v) = \frac{\beta + \sum_{w \in \overrightarrow{\Gamma}(u) \cap \overrightarrow{\Gamma}(v)} \overrightarrow{\mathtt{A}_{\mathcal{G}}}(u,v,w)}{\left| \overrightarrow{\Gamma}(u) \cup \overrightarrow{\Gamma}(v) \right|}, \quad \text{where } \beta > 0. \tag{5}$$

Likewise, we define the aggregated agreement of two trustees, as the sum of their agreements divided by the number of all their trustors:

$$\overleftarrow{\mathtt{AGR}_{\mathcal{G}}}(u,v) = \frac{\beta + \sum_{w \in \overleftarrow{\Gamma}(u) \cap \overleftarrow{\Gamma}(v)} \overleftarrow{\mathtt{A}_{\mathcal{G}}}(u,v,w)}{\left| \overleftarrow{\Gamma}(u) \cup \overleftarrow{\Gamma}(v) \right|}, \quad \text{where } \beta > 0. \tag{6}$$

The parameter β in Eq. (5) and Eq. (6) is a *bootstrapping factor*. It serves in not excluding users with no common trustees (Eq. (5)) or no common trustors (Eq. (6)). To clarify its utility, let us suppose that $\beta = 0$. Since we intend to weight trust recommendations by agreement scores, some of q's trustors, and some of p's trustees, may be *excluded* from the recommendation process because they have a null agreement score with p, and with q, respectively.

A simple explanation of the parameter β is given as follows: consider two nodes u and v that share no common trustees. Without β, we would have $\overrightarrow{\mathtt{AGR}_{\mathcal{G}}}(u,v) = 0$ no matter how much trustees they have in total. However, agreement may still be expressed by the inverse of the *missed chances* to share trustees. For example, if u has 3 trustees and v has 2 trustees, then they are more likely to agree than if they had 1000 trustees each, without a single common one.

That is, the more $\left| \overrightarrow{\Gamma}(u) \cup \overrightarrow{\Gamma}(v) \right|$ increases, while the intersection $\overrightarrow{\Gamma}(u) \cap \overrightarrow{\Gamma}(v)$ is still empty, the less u agrees with v. We, thus, add the bootstrapping factor β to include other nodes without shared trustees with u, yet keep them ranked by their agreement with u.

Again, and akin to Eq. (4), we notice that:

$$\overleftarrow{\mathtt{AGR}_{\mathcal{G}}}(w,z) = \overrightarrow{\mathtt{AGR}_{\mathcal{G}^T}}(w,z), \tag{7}$$

3.5. Our algorithm

As described in Algorithm 1, to predict how would a node p trust another node q, we rank the trustors of q based on their agreement with p and calculate the weighted mean of their trusts toward q by their agreement with p. The algorithm returns a tuple containing both the inferred trust (the weighted mean) and a mean of the agreement scores which we consider as confidence measure of the result (The more p agrees with

other trustors of q, the more we expect the result to be correct.)

Algorithm 1: Trust inference by trustors agreement

Data: Graph:$\mathcal{G}(\mathcal{N}, \mathcal{E}, \mathcal{W})$, Trustor: p, Trustee q
Result: Inferred Trust value from p to q and a confidence score.

Function `InferByTrustorsSimilarity`($\mathcal{G}$, p, q):
  `others` $\leftarrow 0$;
  `trust` $\leftarrow 0$;
  `infered` $\leftarrow 0$;
  `confidence` $\leftarrow 0$;
  for $\mathtt{c_i} \in \overleftarrow{\Gamma}(\mathtt{q}) \setminus \{\mathtt{p}\}$ **do**
    `score` $\leftarrow \overrightarrow{\mathtt{AGR}_{\mathcal{G}}}(\mathtt{c_i}, \mathtt{p})$;
    `trust` $\leftarrow \mathcal{W}(\mathtt{c_i}, \mathtt{q}) \times$ `score`;
    `infered` $\leftarrow$ `infered` $+$ `trust`;
    `confidence` $\leftarrow$ `confidence` $+$ `score`;
    `others` $\leftarrow$ `others` $+ 1$;
  end
  if `others` > 0 **then**
    `infered` $\leftarrow$ `trust`/`confidence`;
    `confidence` $\leftarrow$ `confidence`/`others`;
  end
  return $\langle$ `infered`, `confidence` $\rangle$
End Function

Similarly, to predict the trust that a node q should receive from another node p, we rank the trustees of p based on their trustees-agreement with q, and calculate a weighted mean of the trust that p puts in them weighted by their respective agreement score with q. We return a tuple containing the weighted mean as an inferred trust, and the scores mean as a confidence measure of the inference. Luckily, the observation made in Eq. (7), allows us to reuse Algorithm 1 with the transpose of G.

Finally, the inferred trust from p to q using both methods is the average mean of their results weighted by their confidence scores as shown in Algorithm 2.

Algorithm 2: Trust inference by agreement

Data: Graph:$\mathcal{G}(\mathcal{N}, \mathcal{E}, \mathcal{W})$, Trustor: p, Trustee q.
Result: Inferred Trust value from p to q.

Function `InferByAgreement`(G, q, p):
  `trust` $\leftarrow 0$;
  $\langle \mathtt{outTrust}, \mathtt{outScore} \rangle \leftarrow \mathtt{InferByTrustorsSimilarity}(\mathcal{G}, \mathtt{p}, \mathtt{q})$;
  $\langle \mathtt{inTtrust}, \mathtt{inScore} \rangle \leftarrow \mathtt{InferByTrustorsSimilarity}(\mathcal{G}^{\mathsf{T}}, \mathtt{q}, \mathtt{p})$;
  if `outScore` $+$ `inScore` > 0 **then**
    $\mathtt{trust} \leftarrow \dfrac{((\mathtt{inTrust} \times \mathtt{inScore}) + (\mathtt{outTrust} \times \mathtt{outScore}))}{(\mathtt{inScore} + \mathtt{outScore})}$;
  end
  return `trust`;
End Function

The time complexity of our algorithm when trying to predict the trust that a node u would put in another one v is $O(2\big|\overrightarrow{\Gamma}(u)\big|\big|\overleftarrow{\Gamma}(v)\big|)$ since we examine all u's trustees and all v's trustors twice.

4. Experimental evaluation

4.1. Datasets description

Table 2 shows some statistics of the real-world datasets that were used during our various tests. The first three were taken from the *Stanford Large Network Dataset Collection*[1], and the last one from *Trustlet*[2].

Bitcoin Alpha and OTC : being an anonymous cryptocurrency, bitcoin users need to assess the trustworthiness of other users they trade with. This need, exacerbated by the risk of fraudulent users, led to the emergence of several exchanges, where users state the level of trust they put in each other (Kumar et al., 2016). Among these exchanges, we use two datasets created by (Kumar et al., 2016) for two Bitcoin-Exchanges: Bitcoin-Alpha, and Bitcoin-OTC. Trust ratings from these two exchanges, originally in the interval -10 (total distrust) to $+10$ (total trust), were scaled to fit in the interval $[-1, +1]$.

Wikipedia Rfa : Wikipedia administrators are elected by the community members in response to a *Request for adminship* (Rfa) that has been submitted by the user himself/herself or another member (West et al., 2014). Originally, the opinions of the community members about such requests are expressed by a vote ($+1$ positive, 0 neutral, or -1 negative) accompanied by a comment. These comments were analyzed using the VADER sentiment engine (Gilbert, 2014) in order to generate a WSN with weights in $[-1, +1]$ (Kumar et al., 2016).

Advogato : is a social network for developers to rate each other's authority. An advogato user may rate another as an *observer*, an *apprentice*, a *journeyer*, or a *master*. These rating were mapped to real numbers (0.1, 0.4, 0.7, and 0.9 respectively) as done by Yao et al. (2013). The resulting network is, in fact, unsigned but we wanted to evaluate how our algorithm and others perform with such a dataset.

Network	Nodes	Arcs	Description
Bitcoin-Alpha	3783	24186	Trust from bitcoin user p to user q.
Bitcoin-OTC	5881	35592	Trust from bitcoin user p to user q.
Wikipedia Rfa	9654	104554	Support or opposition to the election of a user q as a Wikipedia administrator, by another user p.
Advogato	5417	51327	Trust from Advogato user p to user q.

Table 2. Statistics about the used datasets

4.2. Evaluated algorithms

We conducted a series of inference tests on the forecited datasets using the following algorithms:

[1] http://snap.stanford.edu/data/
[2] http://www.trustlet.org/datasets/

Reciprocal this is the simplest one and relies on the assumption that if a node u trusts another node v, then v will probably trust u back as much as it trusts it. That is, $\mathcal{W}(u,v) = \mathcal{W}(v,u)$ if the arc from v back to u exists, and 0 otherwise.

Bias and Deserve (BAD) we took $\texttt{DESERVE}(v)$ as the inferred trust value as described in Mishra and Bhattacharya (2011).

Fairness-Goodness (FxG) the inferred trust value as proposed in Kumar et al. (2016).

Inference by Agreement our *inference by agreement* algorithm described in Section 3, with $\lambda = 2$ and $\beta = e^{-\lambda}$.

4.2.1. Performance evaluation metrics

Chai and Draxler (2014) have recommend to use a combination of metrics to assess the performance of models. For our study, given x_i and y_i the actual, and inferred trust values respectively for the i^{th} arc, we measured the performance of these algorithms using the following three metrics:

Mean Absolute Error (MAE) is the mean of the absolute differences between the actual trust value and the inferred one:

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

Root Mean Squared Error (RMSE) is the square mean of the absolute square differences between the actual trust value and the inferred one:

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

Pearson Correlation Coefficient (PCC) ranges between -1 and $+1$ and indicates how the predicted trust values correlate with the actual ones. The more the PCC converges towards $+1$, the more the two values are correlated:

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

where $\bar{x}$ and $\bar{y}$ are the arithmetic means of x_i and y_i respectively, and $i = 1 \cdots N$.

4.3. Experiments

To measure the performances of our algorithm, we have conducted two types of experiments. In the first tests series, we have evaluated the accuracy of the studied algorithms in predicting an unknown arc's weight using the rest of the network's arcs. The second test series allows us to see how would these algorithms behave in case of network sparsity, where some or most of the global network is invisible due to privacy or optimization concerns. Details of these experiments are given in what follows.

4.3.1. Leave One-Out prediction

This is a usual inference test for social networks. It consists in removing an arc from the graph, and predicting its weight given the information of the rest of the network.

We carried out various tests on the four datasets using the forecited algorithms on every arc of the datasets, and calculated the MAE, RMSE, and PCC metrics for every combination of algorithm and dataset. Table 3 shows the results of these tests, and clearly demonstrates that our *inference by agreement* outperforms all the other algorithms, on every metric (MAE, RMSE, PCC), and on every dataset.

	Bitcoin-Alpha	**Bitcoin-OTC**	**Wikipedia-Rfa**	**Advogato**
Reciprocal	(**0.12**, 0.27, 0.47)	(0.15, 0.32, 0.46)	(0.27, 0.35, 0.04)	(0.50, 0.60, 0.01)
FxG	(0.19, 0.33; 0.24)	(0.22, 0.38, 0.30)	(0.18, 0.24, 0.42)	(0.17, 0.21, 0.52)
BAD	(0.20, 0.34, 0.24)	(0.23, 0.40, 0.32)	(0.18, 0.23, 0.44)	(0.14, 0.21, 0.48)
Agreement	(0.14, **0.24, 0.56**)	(**0.14, 0.26, 0.69**)	(**0.17, 0.22, 0.53**)	(**0.10, 0.16, 0.76**)

Table 3. Results of the leave One-Out tests. Each cell of this table contains a tuple (MAE, RMSE, PCC) of the results of the algorithm (row) on the dataset (column). Lower MAE and RMSE, and higher PCC are better.

4.3.2. Leave $\mathcal{N}$% Out predictions

In order to study the robustness of our approach to network sparsity, we have conducted a series of tests on all the datasets where we remove $\mathcal{N}\%$ arcs, and try to predict their weights using the remaining ones. Note that we do not set the weights of the removed arcs to 0 to consider them as removed, but we completely ignore their existence. This distinction is important because when the arc's weight is set to 0, it is assumed that the arc exists, and that we are trying to predict its weight, whereas when we discard the arc, we are trying to predict its weight were it to form in the future.

To conduct these tests, we have randomly removed 10% edges, then 20%, and so on, up to 90% edges in steps of 10% and tried to predict their weights. We have repeated the tests 100 times for every percent, and took the averages of the resulting MAE, RMSE, and PCC for every algorithm, on every dataset. Figure 3 shows how networks sparsity affects the four algorithms, and again demonstrates that our *inference by agreement* provides the best prediction and is very robust to network sparsity.

5. Discussion

In a *leave-one-out* setting, and as shown in Table 3, our algorithm is the most accurate one among the studied methods. A special case worth mentioning is that in terms of MAE, we notice that our approach is neck-to-neck with the *Reciprocal* algorithm on the Bitcoin datasets, and this may be attributed to the high reciprocity of trust in these networks. However in terms of RMSE (which penalizes big differences between the inferred and the original values), and PCC, our approach is more accurate than the reciprocal one.

As for the *leave-$\mathcal{N}$%-out* tests, the plots in Fig. 3 show a very slow decrease of PCC and a very slow increase of MAE and RMSE for our algorithm compared to the other methods. In fact, for our approach, the average change that occurs in the evaluation metrics, while going from an almost fully-visible network (10%) to an almost fully-hidden (90%) one, is

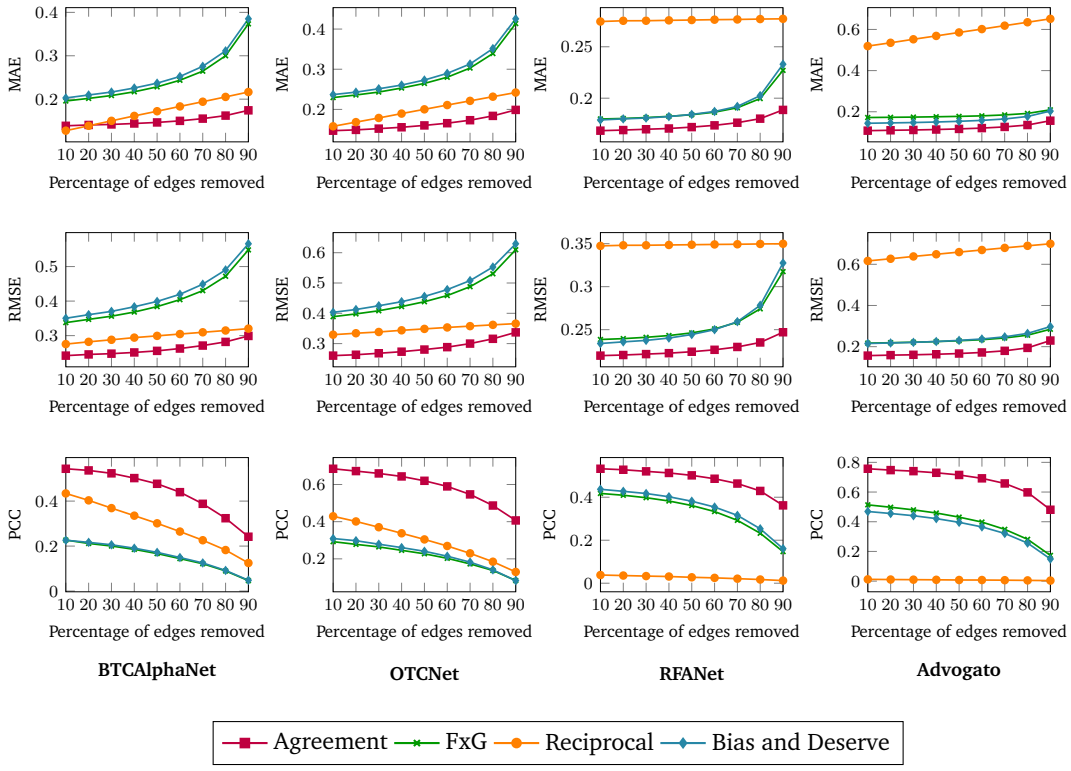


Figure 3. Leave $\mathcal{N}\%$ edges Out results. The x-axis are the percentage of removed edges. The three rows of plots describe how the MAE, RMSE, and PCC respectively change as we remove more edges from the datasets.

about $+0.04$ in MAE, $+0.05$ in RMSE, and -0.25 in PCC. These results clearly demonstrate the robustness of our proposed method to networks sparsity. It may be explained by the way these algorithms operate: global metrics need most, if not all, of the network to be visible to accurately calculate the nodes' characteristics. Whereas a local approach, such as the one we propose, relies only on the direct neighbors of the edge's ends (the trustor and the trustee). Therefore, if the hidden portion of the network does not include that *region*, the prediction will not be affected.

Considering the efficiency of the tested methods, and contrary to the iterative algorithms (FxG and BAD), which are linear, the time complexity for inferring trust from u to v using our approach is $O(2|\overrightarrow{\Gamma}(u)||\overleftarrow{\Gamma}(v)|)$. This time complexity, while non-linear, should generally be acceptable since social networks are arguably not complete graphs. Additionally, one of the advantages of our approach is its locality. Indeed, the two iterative methods rely on global metrics, which should be recalculated on every change that occurs in the networks (new nodes, new edges, update of an edge's weight, etc.). Such events are more than frequent in active social networks. Moreover, knowing that agreement is symmetric, i.e., $\overrightarrow{\texttt{AGR}_{\mathcal{G}}}(u,v) = \overleftarrow{\texttt{AGR}_{\mathcal{G}}}(v,u)$, our implementation has been further

optimized by caching already computed agreement scores.

Finally, and interestingly, we notice that the *Reciprocal* algorithm provides somewhat good results with the two bitcoin datasets, and bad ones with the Wikipedia-Rfa and the advogato ones. This may be attributed to the very nature of the networks. Indeed, the bitcoin networks are about trust in trade, and when there is satisfaction it is generally shared and expressed by both the seller and the buyer, probably out of courtesy if not for anything else. On the other hand, with networks such as the Wikipedia-Rfa and Advogato, what matters most is expertise or authority. An expert being trusted by ordinary trustors does not feel the need, nor is he/she required, to *return the favor*. By considering such a side observation, we can probably improve trust prediction by taking into account the network's own properties.

6. Conclusion and future work

We have explored in this paper the efficiency of *agreement* as a similarity metric for a CF-based trust prediction in weighted signed social networks. Our various experiments have shown that this approach is not only able to provide accurate predictions, but is also very robust to network sparsity. In fact, our approach is barely affected by network sparsity as demonstrated by the *leave-$\mathcal{N}$%-out* tests.

As a future work, we would like to explore the various characteristics of agreement as a concept. Is it transitive? If so, how can we easily deduce the agreement between x and z by knowing the agreement scores between x and y; and between y and z. Also worth extensively studying are the agreement decay and bootstrapping parameters (λ and β respectively). Are they network-dependant, or node-specific? Answers to this sort of questions can enhance even further the speed and the accuracy of agreement calculation; and improve our understanding of social ties and weighted signed networks in general.

Finally, as opposed to most propagation-based methods, we did not use the trust that the target trustor puts in the trustors of the target trustee, we relied only on agreement as a recommendation weight. It may be worth considering as an additional factor that will enable us to enhance the accuracy of our prediction approach.

7. References

Aggarwal, C.C., et al., 2016. Recommender systems. Springer. pp. 8–9. doi:10.1007/978-3-319-29659-3.

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.

Borzymek, P., Sydow, M., Wierzbicki, A., 2009. Enriching trust prediction model in social network with user rating similarity, in: Computational Aspects of Social Networks, 2009. CASON'09. International Conference on, IEEE. pp. 40–47. doi:10.1109/CASoN.2009.30.

Cartwright, D., Harary, F., 1956. Structural balance: a generalization of heider's theory. Psychological review 63, 277.

Chai, T., Draxler, R.R., 2014. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. Geoscientific model development 7, 1247–1250. doi:10.5194/gmd-7-1247-2014.

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.

Cho, J.H., Chan, K., Adali, S., 2015. A survey on trust modeling. ACM Comput. Surv. 48, 28:1–28:40. doi:10.1145/2815595.

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 Inernational 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.

Garakani, M.R., Jalali, M., 2014. A trust prediction approach by using collaborative filtering and computing similarity in social networks, in: Technology, Communication and Knowledge (ICTCK), 2014 International Congress on, IEEE. pp. 1–4. doi:10.1109/ICTCK.2014.7033535.

Ghodousi, E., Hamzeh, A., 2015. A new approach for trust prediction by using collaborative filtering based of pareto dominance in social networks. Ciência e Natura 37, 95. doi:10.5902/2179460x20758.

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.

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.

Hawley, K., 2013. Trust, distrust and commitment. Noûs 48, 1–20. doi:10.1111/nous.12000.

Heider, F., 1946. Attitudes and cognitive organization. The Journal of psychology 21, 107–112.

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 Comput. Surv. 49, 10:1–10:35. doi:10.1145/2906151.

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.

Jones, J.J., Settle, J.E., Bond, R.M., Fariss, C.J., Marlow, C., Fowler, J.H., 2013. Inferring tie strength from online directed behavior. PLOS ONE 8, 1–6. doi:10.1371/journal.pone.0052168.

Jones, K., 1996. Trust as an affective attitude. Ethics 107, 4–25.

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.

Kleinberg, J.M., 1999. Authoritative sources in a hyperlinked environment. Journal of the ACM (JACM) 46, 604–632. doi:10.1145/324133.324140.

Korovaiko, N., Thomo, A., 2013. Trust prediction from user-item ratings. Social Network Analysis and Mining 3, 749–759. doi:10.1007/s13278-013-0122-z.

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.

Leskovec, J., Huttenlocher, D., Kleinberg, J., 2010. Signed networks in social media, in: Proceedings of the SIGCHI conference on human factors in computing systems, ACM. pp. 1361–1370. doi:10.1145/1753326.1753532.

Liu, H., Hu, Z., Mian, A., Tian, H., Zhu, X., 2014. A new user similarity model to improve the accuracy of collaborative filtering. Knowledge-Based Systems 56, 156–166. doi:10.1016/j.knosys.2013.11.006.

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.

Mayer, R.C., Davis, J.H., Schoorman, F.D., 1995. An integrative model of organizational trust. Academy of management review 20, 709–734.

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.

Robbins, B.G., 2016. What is trust? a multidisciplinary review, critique, and synthesis. Sociology Compass 10, 972–986. doi:10.1111/soc4.12391.

Ruan, Y., Durresi, 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.

Sherchan, W., Nepal, S., Paris, C., 2013. A survey of trust in social networks. ACM Comput. Surv. 45, 47:1–47:33. doi:10.1145/2501654.2501661.

Tang, J., Chang, S., Aggarwal, C., Liu, H., 2015. Negative link prediction in social media, in: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, ACM. pp. 87–96. doi:10.1145/2684822.2685295.

Tang, J., Chang, Y., Aggarwal, C., Liu, H., 2016. A survey of signed network mining in social media. ACM Comput. Surv. 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.

Wang, Y., Wang, X., Tang, J., Zuo, W., Cai, G., 2015. Modeling status theory in trust prediction., in: AAAI, pp. 1875–1881.

West, R., Paskov, H.S., Leskovec, J., Potts, C., 2014. Exploiting social network structure for person-to-person sentiment analysis. arXiv preprint arXiv:1409.2450 .

Xiang, R., Neville, J., Rogati, M., 2010. Modeling relationship strength in online social networks, in: Proceedings of the 19th International Conference on World Wide Web, ACM, New York, NY, USA. pp. 981–990. doi:10.1145/1772690.1772790.

Yao, Y., Tong, H., Yan, X., Xu, F., Lu, J., 2013. Matri: a multi-aspect and transitive trust inference model, in: Proceedings of the 22nd international conference on World Wide Web, ACM. pp. 1467–1476. doi:10.1145/2488388.2488516.

Yap, J., Harrigan, N., 2015. Why does everybody hate me? balance, status, and homophily: The triumvirate of signed tie formation. Social Networks 40, 103–122. doi:10.1016/j.socnet.2014.08.002.

Yuan, W., Li, C., Han, G., Guan, D., Zhou, L., He, K., 2017. Negative sign prediction for signed social networks. Future Generation Computer Systems doi:10.1016/j.future.2017.08.037.

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., 2007. Investigating interactions of trust and interest similarity. Decision Support Systems 43, 460–475. doi:10.1016/j.dss.2006.11.003. emerging Issues in Collaborative Commerce.

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.