Strategies for Incorporating Knowledge Defects and Path Length in Trust Aggregation

Nele Verbiest[1], Chris Cornelis[1], Patricia Victor[1] and Enrique Herrera-Viedma[2]

[1] Dept. of Appl. Math. and Comp. Sci., Ghent University, Gent, Belgium
{Nele.Verbiest,Chris.Cornelis,Patricia.Victor}@UGent.be
[2] Dept. of Comp. Sci. and AI, University of Granada, Granada, Spain
viedma@decsai.ugr.es

Abstract. The ability for a user to accurately estimate the amount of trust to be placed in a peer user is gaining more and more attention in social network applications. Trust aggregation provides this ability by identifying paths that connect users in the network, and by merging trust opinions expressed by users along these paths. However, as individual trust opinions are not always based on perfect knowledge, and since the quality of a trust estimation propagated along a given path may diminish as its length increases, mechanisms are needed to handle these imperfections. In this paper, we propose a set of trust aggregation operators that take into account knowledge defects and path length. We investigate their properties, and discuss how they may be implemented in practice, taking into account characteristics of the network such as the availability of a central authority, or the need to preserve users' privacy by not publically disclosing their trust information.

1 Introduction

A trust network consists of agents (users) that can express their opinion about other agents through trust scores. Trust networks are emerging as an important tool to improve web applications such as recommender systems [1–4], e-mail filtering [5], financial transactions [6] and many more.

Generally, not all agents are connected in a trust network. In order to predict an agent's trust score in one of its peers, we can search for a path linking them and propagate trust scores along this path to obtain an estimated trust score. If more than one path is available, we need to aggregate these trust scores to obtain a final estimated trust score. In the aggregation process, not all trust scores may be equally useful, due to the way in which they were obtained. In this paper, we consider two characteristics that affect a trust score's relative importance:

- *Knowledge defects.* Sometimes, an agent has insufficient information to establish a clear trust opinion in a peer, while at other times his opinion is based on conflicting information sources. Both cases are examples of defective knowledge, and can be modelled by drawing trust scores from a bilattice as proposed in [4]. In this way, trust scores not only reflect the intensity of the trust relation, but also the amount of information on which it is based.

- *Path length.* As the number of propagation steps needed to obtain a trust score increases, so does the chance of errors occurring and accumulating in the process. This is also endorsed by the experiments in [2], which show that shorter paths lead to more accurate trust estimations and recommendations.

The remainder of this paper is structured as follows: in Section 2, we recall necessary preliminaries on (ordered) weighted average aggregation operators, the bilattice-based trust model and trust score propagation, while in Section 3, we propose new classes of aggregation operators taking into account knowledge defects and path length. In Section 4, we discuss their application in a trust network; we distinguish between "big brother" approaches in which a central authority performs all propagation and aggregation steps, and distributed ones in which agents process trust information autonomously and do not need to disclose the provenance of this information to their peers. The latter approaches become especially relevant in the light of preserving users' privacy in web-based applications. In Section 5, we conclude and discuss future work.

2 Preliminaries

2.1 Aggregation Operators

In this section, we review a number of frequently used strategies for aggregating a vector $\langle a_1, \ldots, a_n \rangle$ of scalar arguments.

The simple Weighted Average (WA) operator associates weights to the values that have to be aggregated. Let $W = \langle w_1, \ldots, w_n \rangle$ be a weight vector such that $\forall i \in \{1, \ldots, n\}$ $w_i \in [0, 1]$ and $\sum_{i=1}^{n} w_i = 1$. Then the WA-operator associated with W is defined as:

$$WA_W(a_1, \ldots, a_n) = \frac{\sum_{i=1}^{n} w_i a_i}{\sum_{i=1}^{n} w_i} \tag{1}$$

The Ordered Weighted Average (OWA, [7]) operator associates weights to the ordered positions of the values that have to be aggregated. Let $W = \langle w_1, \ldots, w_n \rangle$ be defined as above. Then the OWA-operator associated with W is defined as:

$$OWA_W(a_1, \ldots, a_n) = \sum_{i=1}^{n} w_i b_i \tag{2}$$

where b_i is the i-th largest element in $\{a_1, \ldots, a_n\}$.

The Induced Ordered Weighted Average (IOWA, [8]) operator associates weights to the ordered positions of the values of an order inducing variable. Let $W = \langle w_1, \ldots, w_n \rangle$ be defined as above, and $\langle v_1, \ldots, v_n \rangle$ a vector of values drawn from a linearly ordered space $(V, \leq_V)$. Then the IOWA-operator associated with W is defined as:

$$IOWA_W(\langle a_1, v_1 \rangle, \ldots, \langle a_n, v_n \rangle) = \sum_{i=1}^{n} w_i b_i \tag{3}$$

where $b_i = a_j$ iff v_j is the i-th largest element in $\{v_1, \ldots, v_n\}$.

2.2 Bilattice-Based Trust Network Model

According to the proposal in [4], we model a trust network as a triplet (A, E, R) where (A, E) is a directed graph with the agents as nodes, and directed trust links as edges. The mapping $R : E \to [0, 1]^2$ associates a trust score $R(e) = (t, d)$ with every edge e between agents x and y, such that t represents the degree of trust of x in y, and d the degree of distrust. The set of trust scores can be endowed with the following bilattice structure:

$$BL^{\square} = ([0, 1]^2, \leq_t, \leq_k, \neg) \tag{4}$$

where $\leq_t$ is a trust ordering, $\leq_k$ is a knowledge ordering and $\neg$ a negation operator, defined as follows:

$$\begin{aligned} (t_1, d_1) \leq_t (t_2, d_2) &\text{ iff } t_1 \leq t_2 \text{ and } d_1 \geq d_2 \\ (t_1, d_1) \leq_k (t_2, d_2) &\text{ iff } t_1 \leq t_2 \text{ and } d_1 \leq d_2 \\ \neg(t_1, d_1) &= (d_1, t_1) \end{aligned}$$

$\leq_t$ orders trust scores ranging from full distrust $(0, 1)$ to full trust $(1, 0)$. $\leq_k$ orders trust scores ranging from ignorance/no knowledge $(0, 0)$ to fully conflicting knowledge $(1, 1)$. Trust scores (t, d) with $t + d < 1$ are called *incomplete*, while those with $t + d > 1$ are called *inconsistent*. In both cases, there is a *knowledge defect*, which can be quantified by the following $[0, 1]$-valued measure:

$$kd(t, d) = |1 - t - d| \tag{5}$$

Trust scores with $kd(t, d) = 0$, i.e., $t + d = 1$, are said to have *perfect knowledge*. When explicit trust and distrust values in $[0, 1]$ are given for every trust relation in the network, these can be used for t and d. Otherwise, t and d should be derived from the given numerical or linguistic values that represent trust statements in the given trust network.

2.3 Trust Score Propagation

In trust networks, propagation operators are used to handle the problem of estimating a trust score in an unknown agent by inquiring through third party agents. For our purposes, a trust score propagation operator *Prop* is a $([0, 1]^2)^2 \to [0, 1]^2$ mapping; given the trust score of agent x in agent y, and the trust score of y in z, it predicts the trust score of x in z. Trust score propagation operators have been studied extensively in [4]; here we mention just one example of such an operator, defined by, for $(t_1, d_1), (t_2, d_2)$ in $[0, 1]^2$,

$$Prop((t_1, d_1), (t_2, d_2)) = (\min(t_1, t_2), \min(t_1, d_2)) \tag{6}$$

This operator reflects the basic strategy of taking over information only from trusted sources: it copies the opinion of y in z, but only to the extent that x trusts y.

Although *Prop* defined by (6) is associative, this is not required in general. For propagation over paths with more than 2 edges, it is therefore necessary to fix the evaluation order. Here, as in [9], we assume that a right-to-left evaluation order is used for propagation, illustrated below for a three-edge path:

$$\mathit{Prop}((t_1,d_1),(t_2,d_2),(t_3,d_3)) = \mathit{Prop}((t_1,d_1),\mathit{Prop}((t_2,d_2),(t_3,d_3))) \qquad (7)$$

With this order, each agent along the path combines its trust score in its successor with the propagated trust score it receives from this successor. In this way, an agent needs to have access only to the trust scores it has issued, and propagation does not require the intervention of a central authority.

3 Trust Score Aggregation

In general, a trust score aggregation operator is a mapping $\Omega : ([0,1]^2)^n \to [0,1]^2$ $(n \geq 1)$, where the arguments are trust scores as defined in Section 2.2. Note that the operators from Section 2.1 require scalar values as arguments. As such, they are not directly applicable to aggregate a set of trust scores. Therefore, in accordance with the proposal from [9], we propose trust score aggregation operators that consist of two separate operators, one applied to the trust component and one to the distrust component. Before presenting their definition, we list a number of desirable properties that a trust score aggregation operator should ideally satisfy, based on the recommendations in [9].

3.1 Properties

A trust score aggregation operator Ω may satisfy the following properties:

1. *Idempotence* (**RQ1**). For (t,d) in $[0,1]^2$,

$$\Omega((t,d),\ldots,(t,d)) = (t,d) \qquad (8)$$

2. *Knowledge-invariant trust monotonicity*[3] (**RQ2**). For $(t_i,d_i), (t'_i,d'_i)$ in $[0,1]^2$, if $(t_i,d_i) \leq_t (t^{'}_i,d^{'}_i)$ and $t_i + d_i = t^{'}_i + d^{'}_i$ $(i = 1,\ldots,n)$, then

$$\Omega((t_1,d_1),\ldots,(t_n,d_n)) \leq_t \Omega((t'_1,d'_1),\ldots,(t'_n,d'_n)) \qquad (9)$$

3. *Commutativity* (**RQ3**). If π is a permutation of $\{1,\ldots,n\}$, then for (t_i,d_i) in $[0,1]^2$ $(i = 1,\ldots,n)$,

$$\Omega((t_1,d_1),\ldots,(t_n,d_n)) = \Omega((t_{\pi_1},d_{\pi_1}),\ldots,(t_{\pi_n},d_{\pi_n})) \qquad (10)$$

4. *Neutral element* $(0,0)$ (**RQ4**). For (t_i,d_i) in $[0,1]^2$ $(i = 1,\ldots,n-1)$,

$$\Omega((t_1,d_1),\ldots,(t_{n-1},d_{n-1}),(0,0)) = \Omega((t_1,d_1),\ldots,(t_{n-1},d_{n-1})) \qquad (11)$$

[3] The original property (**RQ2**) from [9] required monotonicity of Ω w.r.t. both $\leq_t$ and $\leq_k$, but is restricted here to monotonicity of $\leq_t$ in case the amount of knowledge in the trust scores remains invariant; monotonicity of $\leq_k$ then becomes trivial.

5. *Opposite arguments* (**RQ5**).

$$\Omega(\underbrace{(1,0),\ldots,(1,0)}_{n \text{ times}},\underbrace{(0,1),\ldots,(0,1)}_{n \text{ times}}) = (1,1) \tag{12}$$

3.2 Aggregation Operators

Below, we discuss various strategies for trust aggregation based on WA, OWA and IOWA operators. We first describe the generic procedure for their application:

1. Given are n trust scores $\langle(t_1,d_1),\ldots,(t_n,d_n)\rangle$. Determine m, the number of arguments (t_i,d_i) $(i=1,\ldots,n)$ such that $kd(t_i,d_i)<1$. If $kd(t_i,d_i)=1$, the trust score does not take part in the remainder of the aggregation process.
2. If $m=0$, the aggregation process terminates at this step, and the final result is set to $(1,1)$ if at least one of the arguments equals $(1,1)$, and to $(0,0)$ otherwise.
3. Renumber the remaining m trust scores to $\langle(t_1,d_1),\ldots,(t_m,d_m)\rangle$.
4. With the same conditions as in Section 2.1, determine weight vectors $W=\langle w_1,\ldots,w_m\rangle$ and $W'=\langle w'_1,\ldots,w'_m\rangle$ and, in the case of an IOWA operator, vectors $\langle v_1,\ldots,v_m\rangle$ and $\langle v'_1,\ldots,v'_m\rangle$ of values for an order inducing variable.
5. The final aggregated trust score equals, for WA, OWA and IOWA, respectively:

$$\begin{aligned}
WA_{W,W'}((t_1,d_1),\ldots,(t_m,d_m)) &= (WA_W(t_1,\ldots,t_m),WA_{W'}(d_1,\ldots,d_m))\\
OWA_{W,W'}((t_1,d_1),\ldots,(t_m,d_m)) &= (OWA_W(t_1,\ldots,t_m),OWA_{W'}(d_1,\ldots,d_m))\\
IOWA_{W,W'}\begin{pmatrix}(\langle t_1,v_1\rangle,\langle d_1,v'_1\rangle),\\ \ldots,\\ (\langle t_m,v_m\rangle,\langle d_m,v'_m\rangle)\end{pmatrix} &= \begin{matrix}(IOWA_W(\langle t_1,v_1\rangle,\ldots,\langle t_m,v_m\rangle),\\ IOWA_{W'}(\langle d_1,v'_1\rangle,\ldots,\langle d_m,v'_m\rangle))\end{matrix}
\end{aligned}$$

The principle of eliminating all $(0,0)$ and $(1,1)$ scores beforehand corresponds to the intuition that these trust scores do not contribute any usable information, because they represent total ignorance and complete inconsistency, respectively, and should thus not participate to the aggregation process.

Weighted Average Trust Score Aggregation. As stated in the introduction, trust scores that exhibit a high knowledge defect (too little, or too much knowledge), are considered less useful in deriving a final trust estimation. In order to penalize this knowledge defect, we can construct weight vectors $W=W'$, such that, for $i=1,\ldots,m$,

$$w_i = \frac{1-kd(t_i,d_i)}{\sum_{j=1}^{m} 1-kd(t_j,d_j)} \tag{13}$$

It is also possible to construct weights based on the path length associated with a trust score. In this scenario, we additionally need the vector $\langle p_1,\ldots,p_m\rangle$, such that p_i is the number of propagations (path length) needed to obtain the trust score (t_i, d_i). The longer a trust score's path length, the smaller its associated weight should be. This can be obtained by putting $W = W'$, such that $\forall i, j = 1, \ldots, m$, $p_i w_i = p_j w_j$. That is, for $i = 1, \ldots, m$,

$$w_i = \frac{\frac{1}{p_i}}{\sum_{j=1}^{m} \frac{1}{p_j}} \tag{14}$$

Both weighting strategies can easily be combined to take into account both knowledge defect and path length, i.e., for $i = 1, \ldots, m$,

$$w_i = \frac{\frac{1-kd(t_i,d_i)}{p_i}}{\sum_{j=1}^{m} \frac{1-kd(t_j,d_j)}{p_j}} \tag{15}$$

It may be verified that the resulting $WA_{W,W'}$ have properties (**RQ1**) (on condition that all path lengths are equal), (**RQ2**) (on condition that path lengths remain invariant), as well as (**RQ3**) and (**RQ4**). Only (**RQ5**) does not hold; for instance, using (13), under the premises of this property, the outcome will be $\left(\frac{1}{2}, \frac{1}{2}\right)$ rather than $(1, 1)$. In general, it is difficult to impose this property using a WA operator.

Ordered Weighted Average Trust Score Aggregation. Ordering the trust (resp., distrust) values decreasingly prior to computing their weighted average offers additional flexibility to the aggregation process. As an example, the following weighting strategy generalizes the proposal made in [9]; for $i = 1, \ldots, m$,

$$w_i = \frac{\max(0, \lceil \frac{m}{m_t} \rceil - i + 1)}{\frac{\lceil \frac{m}{m_t} \rceil (\lceil \frac{m}{m_t} \rceil + 1)}{2}}, w'_i = \frac{\max(0, \lceil \frac{m}{m_d} \rceil - i + 1)}{\frac{\lceil \frac{m}{m_d} \rceil (\lceil \frac{m}{m_d} \rceil + 1)}{2}} \tag{16}$$

with $m_t, m_d \in \{2, \ldots, m\}$. For instance, if $m_t = 2$, it means that half of the trust degrees receive a strictly positive weight. Because of the way the weights are constructed, the weight for the highest trust degree will be strictly greater than the second one, etc. Different values for m_t and m_d may be chosen; for instance, if $m_t < m_d$, the distrust degree of the aggregated trust score will depend on fewer arguments than the trust degree. Using these weights, $OWA_{W,W'}$ satisfies all properties from Section 3.1, including (**RQ5**).

It is also possible to combine the WA weighting strategies (13)–(15) with OWA weights. In this case it is important to realize that the WA weights are associated to a fixed argument, and the OWA weights to an ordered position. In particular, let w_i^{OWA} and ${w'}_i^{OWA}$ be weights defined as in (16) and $w_{\pi_i}^{WA}$ and ${w'}_{\pi_i}^{WA}$ weights defined as in (13)–(15), where π (resp. π') permutes the weights in a way that $t_{\pi_1} \geq t_{\pi_2} \geq \ldots \geq t_{\pi_n}$ (resp. $d_{\pi'_1} \geq d_{\pi'_2} \geq \ldots \geq d_{\pi'_n}$). Then combined weights can be calculated as:

$$w_i = \frac{w_i^{OWA} w_{\pi_i}^{WA}}{\sum_{j=1}^{n} w_j^{OWA} w_{\pi_j}^{WA}}, w'_i = \frac{{w'}_i^{OWA} {w'}_{\pi'_i}^{WA}}{\sum_{j=1}^{n} {w'}_j^{OWA} {w'}_{\pi'_j}^{WA}} \tag{17}$$

Using these combined weights, however, $OWA_{W,W'}$ no longer satisfies (**RQ2**).

Induced Ordered Weighted Average Trust Score Aggregation. Since the order inducing variable appearing in the IOWA approach does not need to be scalar, further flexibility can be introduced in the trust aggregation process. In particular, we may consider combinations of trust/distrust degrees, path length and/or knowledge defect as potential order inducing variables, provided a suitable linear order can be imposed on them.

As a first example, we define the value of the order inducing variables as

$$v_i = (kd(t_i, d_i), t_i),\ \ v_i' = (kd(t_i, d_i), d_i), \tag{18}$$

$i = 1, \ldots, m$, and order these values decreasingly according to the linear order $\leq_{kd}$ on $[0,1]^2$ defined by, for $(k_1, r_1), (k_2, r_2)$ in $[0,1]^2$,

$$(k_1, r_1) \leq_{kd} (k_2, r_2) \Leftrightarrow (k_1 > k_2) \vee (k_1 = k_2 \wedge r_1 \leq r_2) \tag{19}$$

In other words, trust scores with lower knowledge defects are ordered first, and in case of equal knowledge defect, the higher trust (resp., distrust) value prevails.

Another option is to define $v_i = (p_i, t_i)$ and $v_i' = (p_i, d_i)$, $i = 1, \ldots, m$, and order these values decreasingly according to the linear order $\leq_p$ on $\mathbb{N} \times [0,1]$ defined by, for $(p_1, r_1), (p_2, r_2)$ in $\mathbb{N} \times [0,1]$,

$$(p_1, r_1) \leq_p (p_2, r_2) \Leftrightarrow (p_1 > p_2) \vee (p_1 = p_2 \wedge r_1 \leq r_2) \tag{20}$$

In this case, trust scores associated to the shortest path are ordered first, and the trust/distrust degrees are used to determine the order in case of equal path length.

Weight vectors W, W' can be constructed in the same way as for the OWA operators, with the same properties fulfilled. For instance, using (16), trust scores with the highest associated value for the order inducing variable get the highest weights. It is also possible to consider combined approaches that take into account both knowledge defect and path length, but as we have seen before, the combination of (16) with (13)–(15) results in the loss of (**RQ2**).

Instead, we may incorporate path length and knowledge defect simultaneously using alternative order inducing variables $v_i = (p_i kd(t_i, d_i), t_i)$ and $v_i' = (p_i kd(t_i, d_i), d_i)$, and ordering their values decreasingly according to the linear order $\leq_{p,kd}$ on $\mathbb{R}^+ \times [0,1]$ defined as follows for $(s_1, r_1), (s_2, r_2)$ in $\mathbb{R}^+ \times [0,1]$:

$$(s_1, r_1) \leq_{p,kd} (s_2, r_2) \Leftrightarrow (s_1 > s_2) \vee (s_1 = s_2 \wedge r_1 \leq r_2) \tag{21}$$

Using these order inducing variables, and weights W, W' defined by (16), we can verify that $IOWA_{W,W'}$ fulfils all requirements in Section 3.1.

4 Application to Trust Networks

In this section, we discuss how the proposed aggregation operators can be used within a trust network (A, E, R) as defined in Section 2.2 to estimate trust scores.

Assume a and x are two agents in A such that there is at least one path from a to x, but no directed edge from a to x. Examples of this situation are given in Figure 1. We now consider two strategies to predict the trust score of a in x.

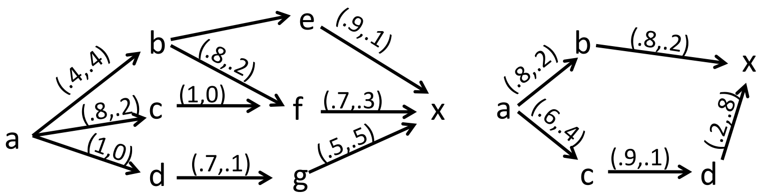

Fig. 1. Examples of trust networks

4.1 Aggregation With a Central Authority ("Big Brother")

In the first approach, all acyclic paths between a and x are considered. For every path, a propagated trust score is calculated, and at the end, all of these propagated trust scores are aggregated. This approach is called a *Big Brother* approach, since a central authority that maintains all trust information (network topology and trust scores) is needed. Any of the aggregation operators introduced in Section 3 can be used to compute the final estimated trust score.

Example 1. In the left hand side network in Figure 1, there are four a-x paths, all of them of length 3 ($abex$, $abfx$, $acfx$ and $adgx$). If we use Eq. (6) to propagate, then the trust scores corresponding to these paths are $(0.4, 0.1)$, $(0.4, 0.3)$, $(0.7, 0.3)$ and $(0.5, 0.5)$, with knowledge defects 0.5, 0.3, 0 and 0, respectively. Constructing the weight vectors $W_1 = W_1'$ by Eq. (13), we get

$$WA_{W_1,W_1'}((0.4, 0.1), (0.4, 0.3), (0.7, 0.3), (0.5, 0.5)) = (0.525, 0.331) \qquad (22)$$

If we construct weights W_2 and W_2' as in (16) with $m_t = 2, m_d = 4$ we get

$$OWA_{W_2,W_2'}((0.4, 0.1), (0.4, 0.3), (0.7, 0.3), (0.5, 0.5)) = (0.633, 0.5) \qquad (23)$$

The same result is obtained if we combine these weights with those in (13) using (17) , and also if we use W_2 and W_2' in the IOWA strategy where the order inducing variables v_i and v_i' are defined as in (18).

4.2 Aggregation Without a Central Authority

If a central authority is lacking, a distributed strategy may be pursued in which agent a asks all agents it is connected to for their opinion of x. Whenever a successor s is directly connected to x, it returns the corresponding trust score to a. Otherwise, it returns a recursively calculated trust score that represents its opinion about x. Agent a then applies propagation to its own trust score in s, and the trust score supplied by s. Finally, a aggregates the trust scores established for its various successors. This is an example of a privacy-preserving approach, since the agents do not have to reveal their trust opinions to a central authority. In this case, aggregation operators incorporating path length may not be suitable, since they require the path length to be transferred along with the trust score as an extra parameter (additional overhead), and moreover this information might be confidential (privacy breach).

On the other hand, if we want to penalize trust scores based on the amount of nodes they have passed through in the trust network, we can apply a so-called ageing function after every propagation step. For example, given a parameter α ($0 < \alpha < 1$) and a trust score (t, d), we can define the ageing function f_α by

$$f_\alpha(t,d) = \begin{cases} (\alpha t, \alpha d) & \text{if } t + d \leq 1; \\ (1 - \alpha(1-t), 1 - \alpha(1-d)) & \text{else.} \end{cases} \quad (24)$$

It is clear that the ageing function increases the knowledge defect; if we subsequently aggregate the results using an aggregation operator that incorporates knowledge defect, trust scores that have come a longer way through the network will have a smaller associated weight.

Example 2. In the right hand side network of Figure 1, a asks c and b for an opinion of x; b responds by returning its trust score in x, to which a applies propagation and ageing (assume $\alpha = 0.9$), i.e., $f_{0.9}(\mathit{Prop}(0.8, 0.2), (0.8.0.2)) = (0.72, 0.18)$. Agent c applies the same operation to the score it gets from d, i.e., it returns $(0.18, 0.72)$ to a; by another propagation and ageing step, a transforms this score to $(0.162, 0.54)$. If we construct weights $W = W'$ as in (13), we obtain

$$WA_{W,W'}((0.72, 0.18), (0.162, 0.54)) = (0.475, 0.338). \quad (25)$$

5 Conclusion and Future Work

In this paper, we have presented and investigated a range of strategies that can be used to aggregate trust scores, and which take into account aspects of the trust network such as the relative knowledge defect of trust scores, and the length of the path that is needed to obtain them. We have also investigated their properties and illustrated their application within a trust network.

In a next step, we plan to validate the introduced aggregation operators experimentally. In particular, we will set up a series of leave-one-out experiments in which a particular edge in the trust network is hidden, and the goal is to predict the corresponding trust score using the remaining connections. On the other hand, we also want to establish the usefulness of the proposed aggregation operators in recommender systems; in this case, the use of a given operator is measured by means of the accuracy it achieves in predicting a user's ratings for given items.

As another part of our future work, we plan to further refine the aggregation strategy to take into account other aspects of the trust network. One possible extension includes the incorporation of an authority function, which reflects that in obtaining a trust score, some agents are always more important than others (e.g., because they have built up a strong reputation).

Acknowledgments

Chris Cornelis would like to thank the Research Foundation—Flanders for funding his research. Patricia Victor would like to thank the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) for funding her research. Enrique Herrera-Viedma would like to thank the financing of andalucian excellence project TIC05299, Feder Funds in FUZZYLING project (TIN2007-61079) and PETRI project (PET 2007-0460).

References

1. P Bedi, H Kaur, and S Marwaha. Trust based recommender system for the semantic web. In *Proc. of IJCAI07*, pages 2677–2682, 2007.
2. J Golbeck. *Computing and applying trust in web-based social networks.* PhD thesis, 2005.
3. P Massa and A Avesani. Trust-aware recommender systems. In *Proc. of the ACM Recommender Systems Conference*, pages 17–24, 2007.
4. P Victor, C Cornelis, M De Cock, and P Pinheiro da Silva. Gradual trust and distrust in recommender systems. *Fuzzy Sets and Systems*, 160(10):1367–1382, 2009.
5. J Golbeck and J Hendler. Reputation network analysis for email filtering. In *Proceedings of the First Conference on Email and Anti-Spam*, 2004.
6. A Josang. Trust-based decision making for electronic transactions. In *Proceedings of the Fourth Nordic Workshop on Secure IT Systems*, 1999.
7. RR Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man, and Cybernetics*, 18:183–190, 1988.
8. RR Yager and D Filev. Induced ordered weighted averaging operators. *IEEE Transactions on Systems, Man, and Cybernetics*, 29(2):141–150, 1999.
9. P Victor, C Cornelis, M De Cock, and E Herrera-Viedma. Aggregation of gradual trust and distrust. In *Proceedings of EUROFUSE 09 (Preference Modelling and Decision Analysis)*, pages 259–264, 2009.