Practical Aggregation Operators for Gradual Trust and Distrust

Patricia Victor[a,*], Chris Cornelis[a], Martine De Cock[b,1], Enrique Herrera-Viedma[c]

[a] *Dept. of Applied Mathematics and Computer Science, Ghent University, Belgium*
[b] *Institute of Technology, University of Washington Tacoma, USA*
[c] *Dept. of Computer Science and Artificial Intelligence, University of Granada, Spain*

Abstract

Trust and distrust are two increasingly important metrics in social networks, reflecting users' attitudes and relationships towards each other. In this paper, we study the indirect derivation of these metrics' values for users that do not know each other, but are connected through the network. In particular, we study bilattice-based aggregation approaches and investigate how they can be improved by using ordered weighted averaging techniques, or through the incorporation of knowledge defects. Experiments on a real world data set from CouchSurfing.org demonstrate that the best operators from a theoretical perspective are not always the most suitable ones in practice, and that the sophisticated aggregation methods can outperform the more obvious bilattice-based approaches.

Keywords: aggregation operator, trust network, distrust, social network, ordered weighted average, bilattice

1. Introduction

The last years have witnessed a proliferation of web applications in which people can share information and interact with each other. These are the

*Corresponding author.
Email addresses: Patricia.Victor@UGent.be (Patricia Victor), Chris.Cornelis@UGent.be (Chris Cornelis), mdecock@u.washington.edu (Martine De Cock), viedma@decsai.ugr.es (Enrique Herrera-Viedma)
[1] On leave from Ghent University.

so-called social networking sites, think e.g. of Facebook[2] (social utility that connects people to friends and others), LinkedIn[2] (business networking tool), Flickr[2] (photo management and sharing) or Wikipedia[2] (free online encyclopedia). A lot of these social networking applications rely heavily on the opinions that their users express: recommendation systems would not perform well if their users did not have the possibility to indicate which products they (dis)like, online marketplaces would lose a lot of their attraction without the feedback option that monitors the behavior of buyers and sellers, question answering systems would not be very popular if there were no evaluation facilities which make it much easier to filter out the best answers, and so on.

Besides opinions on objects, very often *social web applications* allow people to express their view on other users of the system too. Such opinions come in many flavors: users can add their connections as 'friends' in Facebook, bookmark 'interesting people' in Amazon.com[3], allow 'fans' and 'followers' in Yahoo!Answers and Twitter[3], etc. Apart from these positive labels, in a large group of users, each with their own intentions, tastes and opinions, it is only natural that also negative evaluation concepts are needed. For example, the technology news web site Slashdot[4] lets its users tag each other as 'friends', 'fans', 'foes' or 'freaks', and the political forum Essembly[4] as 'friends', 'allies' or 'nemeses'.

In this paper, we focus on social networks in which the users explicitly express their opinion as trust and distrust statements. We refer to this kind of social networks as *trust networks*. An interesting example is CouchSurfing[4], a large worldwide hospitality exchange network. Users can create a profile and indicate if they are offering sleeping accomodation; other users looking for a couch can then browse through the profiles and try to determine which users are trustworthy enough to be their host (and vice versa). To this aim, CouchSurfing provides several evaluation possibilities, such as leaving references or creating friendship relations. After a couch experience, users can also indicate how much they trust or distrust each other, which gives rise to a large (hidden) trust network among the CouchSurfers. Forming your own opinion on the users might have been easy when the network was still rather

[2]See `www.facebook.com`, `www.linkedin.com`, `www.flickr.com` and `wikipedia.org`

[3]See `www.amazon.com`, `answers.yahoo.com`, `twitter.com`

[4]See `slashdot.org`, `www.essembly.com`, `www.couchsurfing.org`

small, but nowadays CouchSurfing contains over one million users, making it increasingly difficult to find the hosts/guests that you would get along with well, let alone the ones that are trustworthy.

In large networks such as CouchSurfing, it is very unlikely that all users know each other directly. In other words, these networks are unlikely to be fully connected. This means that, if a user a wants to form a trust opinion about an unknown user x, a has to inquire about x with one of its own trust relations, say b, who in turn might consult a trust connection, etc., until a user connected to x is reached. The process of predicting the trust score along the thus constructed path from a to x is called *trust propagation*. Since it often happens that a has not one, but several trust connections that it can consult for an opinion on x, we also require a mechanism for combining several trust scores originating from different sources. This process is called *trust aggregation*.

Previously, a number of propagation operators for trust and distrust values have been proposed in [1], where we advocated a trust model in which trust scores are (trust,distrust)-couples drawn from a bilattice [2]. In this paper, we embark on the problem of aggregating information originating from multiple trust paths into an overall score, a research area that is still in its infancy. The problem of trust score aggregation was first addressed in [3] and [4], in which we introduced a number of desirable criteria that a trust score aggregation operator should satisfy, and presented possible aggregation approaches based on the use of Yager's *Ordered Weighted Averaging* (OWA, [5]) operators. In this work, we undertake a more general study of trust score aggregation operators. Furthermore, since individual trust opinions are not always based on perfect knowledge (if often happens that users are not completely certain of their opinion), we also investigate the effect of incorporating *knowledge defects* into the aggregation process.

In Section 2, we first recall preliminaries and introduce new trust score concepts regarding the bilattice-based trust model and its propagation operators. We elaborate on the proposed aggregation criteria in Section 3, present several bilattice-based operators, and discuss their theoretical properties. In Section 4, we introduce weighted extensions of these aggregation operators, with or without taking into account knowledge defects, while in Section 5 we investigate the usefulness of the proposed operators in practice. We examine their applicability on the basis of a data set from CouchSurfing and show how they can be used to improve the less sophisticated aggregation strategies. We

conclude and outline ideas for future investigation in Section 6.

2. Preliminaries

According to the proposal in [1], we model a trust network as a directed graph with the users as nodes, and directed trust links as edges:

Definition 1 (Trust network). *A trust network is a couple (A, R) in which A is the set of users and R is an $A \times A \to [0,1]^2$ mapping that associates to each couple (x, y) of users in A a trust score $R(x, y) = (t, d)$ in $[0,1]^2$.*

Definition 2 (Trust score). *A trust score (t, d) is an element of $[0,1]^2$, in which t is called the trust degree, and d the distrust degree.*

We will use trust scores to compare the degree of trust and distrust a user may have in other users in the network, or to compare the uncertainty that is contained in the trust scores. This information can e.g. be used in the ranking mechanisms of a recommender system, a file-sharing system, and so on; for example by giving preference to recommendations/files from sources that are trusted more, or to opinions that are better informed.

2.1. The Trust Score Space

The set of trust scores can be endowed with a bilattice structure, giving rise to the trust score space, a trust model that allows to compare and preserve information about the provenance of trust scores. For more background information and a comparison with other trust models, we refer to [1].

Definition 3 (Trust score space, Trust-distrust and Knowledge ordering). *The trust score space*

$$\mathcal{BL}^{\Box} = ([0,1]^2, \leq_{td}, \leq_k, \neg)$$

consists of the set $[0,1]^2$ of trust scores, a trust-distrust ordering $\leq_{td}$, a knowledge ordering $\leq_k$, and a negation $\neg$ defined by

$$\begin{array}{rcl} (t_1, d_1) \leq_{td} (t_2, d_2) & \text{iff} & t_1 \leq t_2 \text{ and } d_1 \geq d_2 \\ (t_1, d_1) \leq_k (t_2, d_2) & \text{iff} & t_1 \leq t_2 \text{ and } d_1 \leq d_2 \\ \neg(t_1, d_1) & = & (d_1, t_1) \end{array}$$

for all (t_1, d_1) and (t_2, d_2) in $[0,1]^2$.

One can verify that the structure $\mathcal{BL}^{\Box}$ is a bilattice in the sense of Ginsberg [2], that is, $([0,1]^2, \leq_{td})$ and $([0,1]^2, \leq_k)$ are both lattices and the negation $\neg$ serves to impose a relationship between them:

$$(t_1, d_1) \leq_{td} (t_2, d_2) \Rightarrow \neg(t_1, d_1) \geq_{td} \neg(t_2, d_2)$$

$$(t_1, d_1) \leq_k (t_2, d_2) \Rightarrow \neg(t_1, d_1) \leq_k \neg(t_2, d_2),$$

such that $\neg\neg(t_1, d_1) = (t_1, d_1)$. In other words, $\neg$ is an involution that reverses the $\leq_{td}$-order and preserves the $\leq_k$-order.

Figure 1 shows $\mathcal{BL}^{\Box}$, along with some examples of trust scores. These scores are interpreted as epistemic values: they reflect the imperfect knowledge we have about the actual trust and distrust values, which are complementary. The lattice $([0,1]^2, \leq_{td})$ orders the trust scores going from complete distrust $(0,1)$ to complete trust $(1,0)$. The lattice $([0,1]^2, \leq_k)$ evaluates the amount of available trust evidence, ranging from a "shortage of evidence", $t + d < 1$, to an "excess of evidence", viz. $t + d > 1$.

Note that it is possible that two trust scores cannot be compared in $([0,1]^2, \leq_{td})$ or $([0,1]^2, \leq_k)$, but that it can never occur that they are incomparable in both lattices. Hence, for the remainder of the paper, any two inputs or outputs of an aggregation process can always be compared to each other, either on the content level or the knowledge level.

The boundary values of the $\leq_k$ ordering, $(0,0)$ and $(1,1)$, reflect ignorance, resp. contradiction. We call trust scores (t,d) with $t + d < 1$ 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:

Definition 4 (Knowledge defect, Knowledge defective trust score). *We define the knowledge defect of a trust score (t,d) as $kd(t,d) = |1 - t - d|$. We say that trust scores (t,d) for which $kd(t,d) = 0$, i.e., $t + d = 1$, have perfect knowledge (i.e., there is no uncertainty about the trust value), while all others are called knowledge defective.*

Definition 5 (Consistent, Inconsistent trust score). *We call a trust score (t,d) consistent iff $t + d \leq 1$, and inconsistent otherwise.*

The bottom part of the bilattice (or lower triangle, under the $kd(t,d) = 0$ line) contains the trust scores for which there is some doubt (uncertainty)

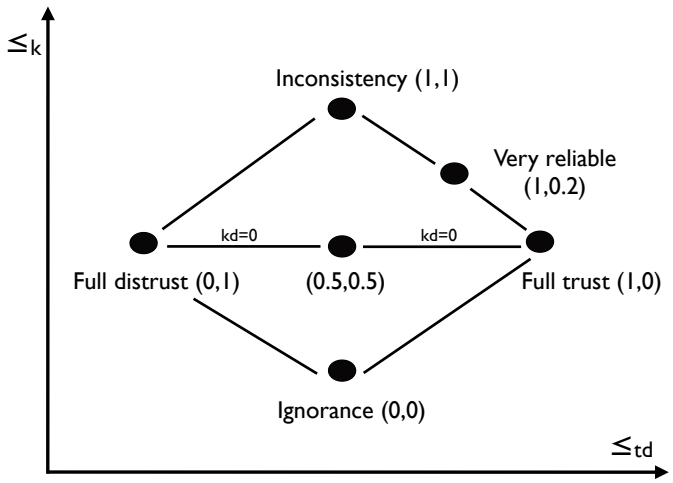

Figure 1: The trust score space $\mathcal{BL}^{\Box}$, a bilattice-based trust model that enables us to compare trust scores according to the trust-distrust content ($\leq_{td}$) and to evaluate the uncertainty that is involved ($\leq_k$).

about the trust degree. The information contained in such a trust score (t, d) can be represented as an interval $[t, 1-d]$, denoting that the user should be trusted at least to the degree t, but not more than $1-d$. In such an interval representation, complete ignorance is represented as $[0, 1]$. Note that this approach is similar to De Cock & Pinheiro da Silva's [6] proposal to model the trust network as an intuitionistic fuzzy relation, and Prade's [7] method where trust evaluations are represented as an interval in a bipolar trust scale.

We call users that issue consistent trust scores consistent users. In this work, we assume that every user is consistent. However, although we start from consistent users, modelling inconsistent information is still needed when we want to accurately represent the result of a trust score aggregation process; we will elaborate upon this in Section 3. The upper part of the bilattice (upper triangle, above the $kd(t, d) = 0$ line) contains such inconsistent trust scores denoting conflicting information. Note that trust scores in the upper triangle cannot be represented as intervals, since they contain too much information instead of a lack.

The trust scores in $\mathcal{BL}^{\Box} = ([0, 1]^2, \leq_{td}, \leq_k, \neg)$ can also be considered within the alternative space $([0, 1]^2, \leq_t, \leq_d, \neg)$, with $\neg$ defined in Definition 3, and

$\leq_t$ and $\leq_d$ as in Definition 6. Note that $\leq_t$ and $\leq_d$ are quasi-orderings, since they are not antisymmetric.

Definition 6 (Trust ordering, Distrust ordering). *The trust ordering $\leq_t$ and distrust ordering $\leq_d$ are defined by*

$$\begin{array}{lll}(t_1, d_1) \leq_t (t_2, d_2) & \text{iff} & t_1 \leq t_2 \\ (t_1, d_1) \leq_d (t_2, d_2) & \text{iff} & d_1 \leq d_2\end{array}$$

The trust and distrust orderings can also be seen as two extra orderings on $\mathcal{BL}^{\Box}$, which separately evaluate the amount of trust and distrust information respectively. The negation $\neg$ serves to impose a relationship between them:

$$(t_1, d_1) \leq_t (t_2, d_2) \Leftrightarrow \neg(t_1, d_1) \leq_d \neg(t_2, d_2).$$

The mapping is illustrated in Figure 2. The dotted line denotes the trust scores (t, d) with perfect knowledge, i.e., $kd(t, d) = 0$ or $t + d = 1$. The triangles underneath (in the gray area) contain the consistent trust scores; inconsistent trust scores reside in the upper triangles.

The bilattice framework allows us to model the trust network as a $\mathcal{BL}^{\Box}$-fuzzy relation in the set of users that associates a score drawn from the trust score space with each ordered pair of users. It should be thought of as a snapshot taken at a certain moment, since trust scores can be updated.

2.2. Trust Score Propagation

In virtual trust networks, propagation operators are used to handle the problem of establishing trust information in an unknown user by inquiring through other users. The simplest case, atomic propagation, takes the trust score of user a in user b and the trust score of b in user x, and uses this information to predict the trust score of a in x. In [1], four operators were proposed for this purpose, each reflecting a different strategy of dealing with the available trust information.

We use $\mathcal{T}$ to denote an arbitrary t–norm, i.e. an increasing, commutative and associative $[0, 1]^2 \to [0, 1]$ mapping satisfying $\mathcal{T}(1, x) = x$ for all x in $[0, 1]$. Furthermore $\mathcal{S}$ denotes an arbitrary t–conorm, i.e. an increasing, commutative and associative $[0, 1]^2 \to [0, 1]$ mapping satisfying $\mathcal{S}(0, x) = x$ for all x in $[0, 1]$. Finally $\mathcal{N}$ is used to denote a negator, i.e. a decreasing $[0, 1] \to [0, 1]$ mapping satisfying $\mathcal{N}(0) = 1$ and $\mathcal{N}(1) = 0$.

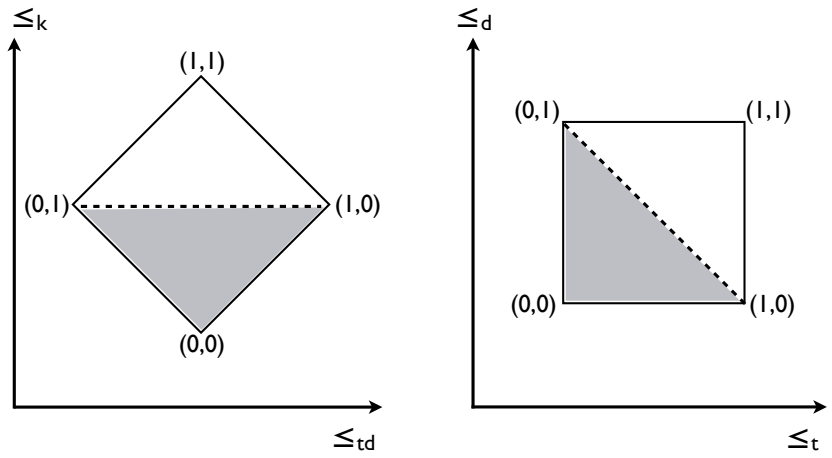

Figure 2: The four orderings on $\mathcal{BL}^{\Box}$: trust-distrust ordering $\leq_{td}$, knowledge ordering $\leq_k$, trust $\leq_t$ and distrust $\leq_d$ ordering.

Definition 7 (Propagation operators in $\mathcal{BL}^{\Box}$). *Let $\mathcal{T}$ be a t-norm, $\mathcal{S}$ a t-conorm and $\mathcal{N}$ a negator. The propagation operators P_1, P_2, P_3 and P_4 are defined by (for (t_1, d_1) and (t_2, d_2) in $[0,1]^2$):*

$$\begin{aligned}
P_1((t_1,d_1),(t_2,d_2)) &= (\mathcal{T}(t_1,t_2),\mathcal{T}(t_1,d_2)) \\
P_2((t_1,d_1),(t_2,d_2)) &= (\mathcal{T}(t_1,t_2),\mathcal{T}(\mathcal{N}(d_1),d_2)) \\
P_3((t_1,d_1),(t_2,d_2)) &= (\mathcal{S}(\mathcal{T}(t_1,t_2),\mathcal{T}(d_1,d_2)),\mathcal{S}(\mathcal{T}(t_1,d_2),\mathcal{T}(d_1,t_2))) \\
P_4((t_1,d_1),(t_2,d_2)) &= (\mathcal{T}(t_1,t_2),\mathcal{S}(\mathcal{T}(t_1,d_2),\mathcal{T}(d_1,t_2)))
\end{aligned}$$

P_1 reflects the basic strategy of taking over information only from trusted sources, while P_2 exhibits a paranoid behavior by taking over distrust information even from an unknown party. P_3, on the other hand, also takes into account distrusted sources, reversing their opinion rather than ignoring or copying it; P_4 mitigates P_3's behavior by reversing only distrust information and ignoring trust information coming from a distrusted party.

Note that only P_1 is associative. The fact that the other operators are not associative means that we need to fix a particular evaluation order to propagate trust scores over paths with more than two edges. In a network with a central authority that maintains all trust information, one can chose which order to use. On the other hand, if there is no central authority, and

each user has access only to the trust scores it has issued, it is necessary to perform the propagation in a right-to-left direction (i.e., right associative propagation operators). With this order, at each node in the propagation path, a user combines its trust score in its successor, with the propagated trust score it receives from this successor. This is illustrated below for a path containing three edges, and a generic propagation operator P:

$$P((t_1, d_1), (t_2, d_2), (t_3, d_3)) = P((t_1, d_1), P((t_2, d_2), (t_3, d_3)))$$

In the remainder, we assume there is a central authority, that the right-to-left evaluation order is used for propagation, and that aggregation is applied at the end of the trust estimation process. In other words, the central authority first looks for paths to the target user, then propagates backwards along these paths, and finally aggregates the values into one final trust score.

Since we assume all users to be consistent, i.e., to issue consistent trust scores, it is desirable that the propagation operators preserve consistency. In this way, we can ensure that all inputs (either direct trust scores or the result of propagations) for the aggregation process are consistent.

Definition 8 (Consistency preserving propagation). *We say that the propagation operator P preserves the consistency iff, when all inputs are consistent, the result of the propagation is consistent too.*

Note that only P_1 and P_2 are consistency preserving for arbitrary choices of the fuzzy logical operators involved; however, when using the minimum/maximum, product/probabilistic sum, or the Łukasiewicz duals, the property also holds for P_3 and P_4.

3. Trust Score Aggregation

When a user a needs to establish an opinion about another user x, and there is more than one path linking them, we require a way of combining the information provided by each of those paths. This process is called aggregation. Trust metrics that only take into account trust mostly use classical aggregation operators such as the minimum, maximum, weighted sum, average, or weighted average [8, 9, 10, 11, 12, 13, 14]. Aggregation of both trust and distrust has not received much attention so far. Guha et al.'s approach [15] uses matrix multiplications to model the propagation and aggregation process, and needs rounding mechanisms to decide whether a user should trust or

distrust another user. In other words, the two values are merged into one. In our trust score space setting, however, the goal is to treat trust and distrust as two separate concepts throughout the whole aggregation process. This strategy is also followed by Jøsang et al in a subjective logic framework: he proposed three probabilistic aggregation operators, called consensus operators, for the fusion of dependent, independent or partially dependent opinions [16], but they assume equally important users, hence lacking flexibility.

In this section, we focus on the trust score aggregation problem. We first postulate several desirable trust score aggregation properties (Section 3.1) and then set out to define a number of aggregation operators fulfilling these properties (Section 3.2).

3.1. Characterization of Trust Score Aggregation Operators

Let $\mathcal{BL}^{\Box} = ([0,1]^2, \leq_{td}, \leq_k, \neg)$ be the trust score space introduced in Section 2.1. In this space, we look for a trust score aggregation operator $A : ([0,1]^2)^n \to [0,1]^2$ $(n \geq 1)$ satisfying as many of the following characteristics pinned down in Definitions 9–17 as possible.

Definition 9 (Trust boundary preservation). *We say that aggregation operator A satisfies the trust boundaries iff, $\forall (t_1, d_1), \cdots, (t_n, d_n) \in [0,1]^2$, $A((t_1, d_1), \ldots, (t_n, d_n)) = (p, q)$, with $\min(t_1, \ldots, t_n) \leq p \leq \max(t_1, \ldots, t_n)$.*

Definition 10 (Distrust boundary preservation). *We say that aggregation operator A satisfies the distrust boundaries iff, $\forall (t_1, d_1), \cdots, (t_n, d_n) \in [0,1]^2$, $A((t_1, d_1), \ldots, (t_n, d_n)) = (p, q)$, with $\min(d_1, \ldots, d_n) \leq q \leq \max(d_1, \ldots, d_n)$.*

Since an aggregated trust score should reflect a consensus about the trust estimation, it is only natural that it should not contain more trust than the maximum trust value among the aggregates. In the same respect, the aggregated distrust value should not be higher than the maximum of the aggregates' distrust values. Analogously, the aggregated trust score should contain at least as much distrust and trust as the minimum among the aggregates.

Note that these conditions imply aggregation operators that cannot be used in an additive context. For instance, in situations where risk is involved, if a lot of the agents highly distrust x, the aggregated distrust degree about x could be experienced on a higher level than the maximum among the distrust degrees, to emphasize the fact that many agents really distrust x (which cannot be modeled due to the upper distrust boundary).

When aggregating additional trust scores, the knowledge contained in the aggregated trust score should not decrease. In other words, the aggregated trust score should contain at least as much knowledge as the most knowledgeable aggregate:

Definition 11 (Knowledge boundary preservation). *We say that an aggregation operator A satisfies the knowledge boundary iff,* $\forall(t_1, d_1), \cdots, (t_n, d_n) \in [0, 1]^2$, $A((t_1, d_1), \ldots, (t_n, d_n)) = (p, q)$, *with* $p + q \geq \max(t_1 + d_1, \ldots, t_n + d_n)$.

Recall that we assume that all users are consistent. If we use propagation operators that preserve consistency, this means that all aggregates will also be consistent, and can be represented as intervals. Hence, in the latter spirit, the knowledge boundary condition implies more narrow intervals, i.e., the uncertainty should not increase.

For example, consider (t_1, d_1) and (t_2, d_2) with $t_1 = t_2$ and $d_1 > d_2$, which means that $[t_1, 1 - d_1]$ is included in $[t_2, 1 - d_2]$. In other words, the latter contains more uncertainty than the former. Due to the trust boundary condition and $t_1 = t_2$, the aggregated trust degree p must be t_1. W.r.t the aggregated distrust degree q, from the knowledge boundary condition ($p + q \geq \max(t_1 + d_1, t_2 + d_2)$) it follows that $q \geq d_1$. Hence, $[p, 1 - q]$ must be at least as narrow as $[t_1, 1 - d_1]$, and certainly less wide than $[t_2, 1 - d_2]$.

Besides these three conditions, the following common properties are also useful, and often imposed on aggregation operators.

Definition 12 (Commutativity, Associativity, Idempotency). *For all* $(t, d), (t_j, d_j) \in [0, 1]^2$ *with* $j \in \{1, \cdots, n\}$, *and for* π *a permutation of* $\{1, \cdots, n\}$, *an aggregation operators A is called*

a) commutative iff $A((t_1, d_1), \ldots, (t_n, d_n)) = A((t_{\pi(1)}, d_{\pi(1)}), \ldots, (t_{\pi(n)}, d_{\pi(n)}))$.

b) idempotent iff $A((t, d), \ldots, (t, d)) = (t, d)$.

c) associative iff $A((t_1, d_1), \cdots, (t_n, d_n)) = A((t_1, d_1), A((t_2, d_2), \cdots, (t_n, d_n)))$ $= \cdots = A(A((t_1, d_1), \cdots, (t_{n-1}, d_{n-1})), (t_n, d_n))$.

Note that, if the trust and distrust boundaries are satisfied, the idempotency condition is automatically fulfilled.

Definition 13 (Trust-distrust monotonicity). *We say that an aggregation operator A respects trust-distrust monotonicity iff* $\forall\ (t_j, d_j), (t_j^{'}, d_j^{'}) \in [0,1]^2$, *with* $j \in \{1, \cdots, n\}$*: if* $(t_j, d_j) \leq_{td} (t_j^{'}, d_j^{'})$, *then*

$$A((t_1, d_1), \ldots, (t_j, d_j), \ldots, (t_n, d_n)) \leq_{td} A((t_1, d_1), \ldots, (t_j^{'}, d_j^{'}), \ldots, (t_n, d_n))$$

Definition 14 (Knowledge monotonicity). *We say that an aggregation operator A respects knowledge monotonicity iff* $\forall\ (t_j, d_j), (t_j^{'}, d_j^{'}) \in [0,1]^2$, *with* $j \in \{1, \cdots, n\}$*: if* $(t_j, d_j) \leq_{k} (t_j^{'}, d_j^{'})$, *then*

$$A((t_1, d_1), \ldots, (t_j, d_j), \ldots, (t_n, d_n)) \leq_{k} A((t_1, d_1), \ldots, (t_j^{'}, d_j^{'}), \ldots, (t_n, d_n))$$

Since we are using a trust score space, with two orderings, two monotonicity conditions arise; one for the trust-distrust and one for the knowledge ordering. Intuitively, each of these conditions makes sense, as for instance, the more information/less doubt (resp., the more trust and the less distrust) the individual sources provide, the more information (resp., the more trust/less distrust) the aggregated outcome should contain. Therefore, a trust score aggregation operator A should be monotonously increasing with respect to both $\leq_{td}$ and $\leq_k$.

Proposition 1. *If A is an idempotent trust score aggregation operator that satisfies the trust-distrust monotonicity, then A respects the trust and distrust boundaries.*

Proof. Due to trust-distrust monotonicity, $A((t_1, d_1), \ldots, (t_j, d_j), \ldots, (t_n, d_n)) \leq_{td} A((t_l, d_m), \ldots, (t_l, d_m))$ [*], with $t_l = \max(t_1, \ldots, t_n)$ and $d_m = \min(d_1, \ldots, d_n)$. Since A is idempotent, [*]$= (t_l, d_m)$. From the definition of $\leq_{td}$ it follows that if $A((t_1, d_1), \ldots, (t_j, d_j), \ldots, (t_n, d_n)) = (p, q)$ then $p \leq t_l \wedge q \geq d_m$, or in other words, $p \leq \max(t_1, \ldots, t_n)$ and $q \geq \min(d_1, \ldots, d_n)$. Analogously for $p \geq \min(t_1, \ldots, t_n)$ and $q \leq \max(d_1, \ldots, d_n)$.

Definition 15 (Trust monotonicity, Distrust monotonicity). *For all* $(t_j, d_j), (t_j^{'}, d_j^{'}) \in [0,1]^2$, *with* $j \in \{1, \cdots, n\}$, *we say that an aggregation operator A respects*

a) trust monotonicity iff A satisfies: if $(t_j, d_j) \leq_t (t_j^{'}, d_j^{'})$, *then* $A((t_1, d_1), \ldots, (t_j, d_j), \ldots, (t_n, d_n)) \leq_t A((t_1, d_1), \ldots, (t_j^{'}, d_j^{'}), \ldots, (t_n, d_n))$.

b) distrust monotonicity iff A satisfies: if $(t_j, d_j) \leq_d (t_j^{'}, d_j^{'})$*, then* $A((t_1, d_1), \ldots, (t_j, d_j), \ldots, (t_n, d_n)) \leq_d A((t_1, d_1), \ldots, (t_j^{'}, d_j^{'}), \ldots, (t_n, d_n))$.

Note that trust-distrust monotonicity is automatically fulfilled if trust and distrust monotonicity hold, because $(p, q) \leq_{td} (r, s) \Leftrightarrow p \leq r \wedge q \geq s \Leftrightarrow (p, q) \leq_t (r, s) \wedge (p, q) \geq_d (r, s)$.

Besides these mathematical properties that have well-known intuitive rationales, we also propose a number of additional requirements to further guarantee the behavior of the trust score aggregation process. We motivate these requirements by examples.

Example 1 (Ignorance). *In the scenario in Figure 3, b and c are both fully trusted acquaintances of a that are connected to x. Propagation with any of the four operators from Section 2.2 results in the two trust scores* (t, d) *and* $(0, 0)$*. However, it can be argued that c's opinion of x (ignorance) should not contribute to the final outcome; indeed, a* $(0, 0)$ *edge can be considered as no edge at all, as it carries no information.*

In other words, $(0, 0)$ should act as a neutral element of the aggregation operator:

Definition 16 (Neutrality). *We say that an aggregation operator A satisfies the neutrality condition iff,* $\forall\ (t_j, d_j) \in [0, 1]^2$ *and* $j \in \{1, \cdots, n-1\}$*,*

$$A((t_1, d_1), \ldots, (t_{i-1}, d_{i-1}), (0, 0), (t_{i+1}, d_{i+1}), \ldots, (t_n, d_n)) = A((t_1, d_1), \ldots, (t_{i-1}, d_{i-1}), (t_{i+1}, d_{i+1}), \ldots, (t_n, d_n)).$$

Example 1 also shows why a naive average of trust and distrust degrees, leading to $\left(\frac{t}{2}, \frac{d}{2}\right)$, would be a poor aggregation strategy in this case.

Proposition 2. *If A is an idempotent trust score aggregation operator that satisfies the knowledge monotonicity and the neutral element condition, then A respects the knowledge boundary.*

Proof. Due to knowledge monotonicity, $A((t_1, d_1), \ldots, (t_i, d_i), \ldots, (t_n, d_n)) \geq_k A((0, 0), \ldots, (t_i, d_i), \ldots, (0, 0))$ [*]. Since A is idempotent and satisfies the neutrality condition, it holds that [*]$= A((t_i, d_i)) = (t_i, d_i)$. Hence, from the definition of $\leq_k$ it follows that if $A((t_1, d_1), \ldots, (t_i, d_i), \ldots, (t_n, d_n)) = (p, q)$ then $p \geq t_i$ and $q \geq d_i$, hence $p + q \geq t_i + d_i$, and this for all $i = 1, \ldots, n$. In other words, $p + q \geq \max(t_1 + d_1, \ldots, t_i + d_i, \ldots, t_n + d_n)$ which shows that A respects the knowledge boundary.

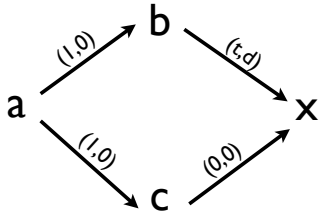

Figure 3: Scenario with ignorance

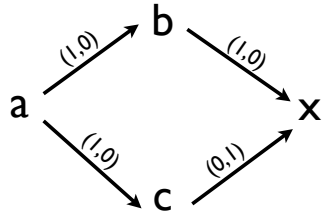

Figure 4: Scenario with total inconsistency

Example 2 (Total inconsistency). *In Figure 4, two fully trusted acquaintances of a express completely opposite trust opinions of x. Again, a simple average of trust and distrust degrees, yielding $(0.5, 0.5)$, is unsuitable, since it does away with the conflicting information a receives, and cannot be distinguished from a scenario in which a receives information that x is half to be trusted and half to be distrusted. In other words, we lose too much provenance information. A more intuitive result seems to be $(1, 1)$, reflecting the inconsistency a faces in its assessment of x.*

This brings us to a final requirement for trust score aggregation operators: an equal number of (1,0) and (0,1) arguments should yield contradiction.

Definition 17 (Opposite arguments). *We say that an aggregation operator A fulfills the opposite arguments condition iff*

$$A(\underbrace{(1,0),\cdots,(1,0)}_{n/2 \text{ times}},\underbrace{(0,1),\cdots,(0,1)}_{n/2 \text{ times}}) = (1,1)$$

3.2. Bilattice-Based Aggregation Operators for Trust Scores

Figure 5 depicts two possible scenarios, both for aggregating two trust scores (denoted by dots): (t_1, d_1) and (t_2, d_2) (Example 3), and (t_3, d_3) and (t_4, d_2) (Example 4). Note that all trust scores are consistent since they reside under or on the $kd = 0$ line (lower triangle); hence, we can also represent them as intervals.

Example 3 (Aggregation with overlap). *User a asks two of his acquintances (whom he trusts completely) for an opinion about user x. The first*

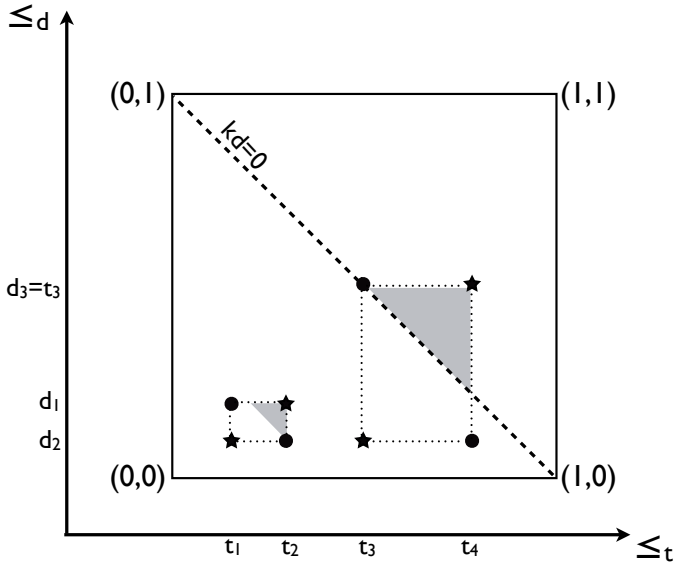

Figure 5: Possible aggregation results of (t_1, d_1) and (t_2, d_2), and (t_3, d_3) and (t_4, d_2), for operators satisfying the trust and distrust boundaries (within the dotted lines) and the knowledge boundary (gray area)

user returns the trust score (t_1, d_1) *and the second one* (t_2, d_2) *(dots). Since user a has complete faith in the two trusted third parties, the propagated trust scores to be aggregated remain* (t_1, d_1) *and* (t_2, d_2) *. Note that, in Figure 5,* $t_1 < t_2$*,* $d_1 > d_2$ *and* $t_2 < 1 - d_1$*; in other words,* $t_1 < t_2 < 1 - d_1 < 1 - d_2$*.*

Since it is desirable that a standard trust score aggregation operator should fulfill at least Definitions 9 and 10 (trust and distrust boundaries), the aggregated result must be in the area marked out by the dotted lines. Hence, the extremes w.r.t. $\leq_t$ and $\leq_d$ for the aggregated trust score are (t_2, d_1) and (t_1, d_2) (stars). By also imposing Definition 11, however, only part of the possible results remain: as can be seen in the figure, $t_2 + d_2 > t_1 + d_1$, and hence all possible results should reside above or on the $kd(t_2, d_2)$ line (formed by all trust scores that have the same knowledge defect as (t_2, d_2)), in other words, in the gray area. In this way, only one extreme remains, namely (t_2, d_1).

The same conclusion can be obtained when reasoning in the interval representation, even without focusing on the boundary conditions. The interval representations of (t_1, d_1) and (t_2, d_2) are $[t_1, 1-d_1]$ and $[t_2, 1-d_2]$ respectively. Note that in Example 3 the two intervals overlap ($t_1 < t_2 < 1 - d_1 < 1 - d_2$). As the aggregated trust interval must reflect the consensus among the two

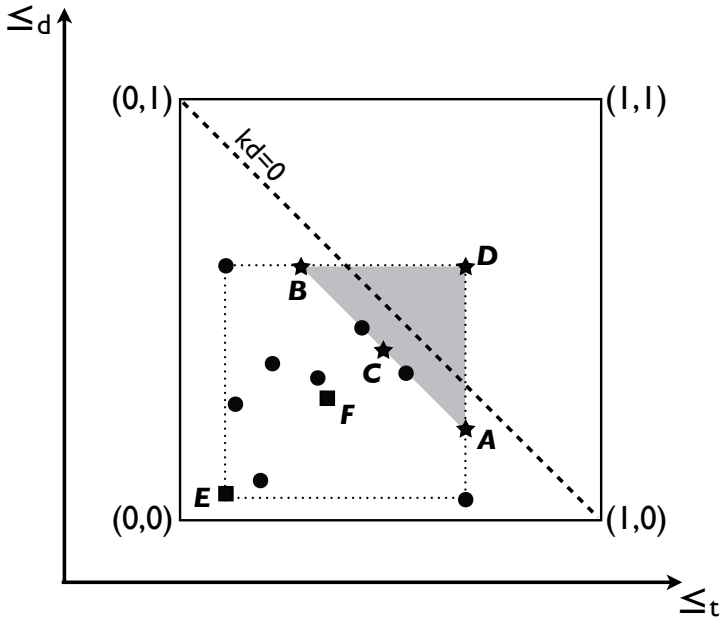

Figure 6: Possible aggregation results of several trust scores, for operators fulfilling the trust, distrust and knowledge boundary conditions

intervals, it should represent the trust estimation on which both users agree, which is usually modeled as the intersection of the two intervals. This strategy results in a narrower interval, or analogously, a trust score that contains less uncertainty (recall our discussion about knowledge monotonicity). Hence, $[t_2, 1-d_1]$ is indeed the most logical choice for the aggregation extreme w.r.t. $\leq_t$ and $\leq_d$.

Example 4 (Aggregation with no overlap). *In the second scenario of Figure 5, user a makes inquiries about user z, resulting in* (t_3, d_3) *(or* $[t_3, 1-d_3]$*) and* (t_4, d_2) *(or* $[t_4, 1-d_2]$*). Note that* $t_3 = 1-d_3 < t_4 < 1-d_2$*. This time, a's acquaintances do not agree at all: there is no overlap between the two trust intervals.*

This example illustrates how inconsistent trust scores can arise, even when the users are consistent. In this case, the extreme aggregated trust score is (t_4, d_3), reflecting the disagreement that exists between the aggregates.

These two-points-examples can be generalized to scenarios with more inputs; see Figure 6 for an example with eight aggregates, represented by dots.

Imposing trust, distrust and knowledge boundary conditions yields a limited number of possible aggregation results, depicted in the figure by the gray area. Each of the trust scores marked by stars makes sense as aggregated trust score: A is the most optimistic choice (maximum trust degree for the lowest possible knowledge level), B the most pessimistic one (maximum distrust degree), C the moderating approach (average of the most knowledgeable trust scores) and D the most extreme, knowledge maximizing, option: maximum trust and distrust degree, often resulting in an inconsistent trust estimation.

We call A and B the trust, resp. distrust maximizing operator, C the knowledge preference averaging operator and D the knowledge maximizing trust score aggregation operator.

Definition 18 (TMAX). *The trust maximizing trust score aggregation operator $TMAX$ is defined as*

$$TMAX\left((t_1,d_1),\cdots,(t_n,d_n)\right) = \\ \left(\max\left(t_1,\cdots,t_n\right),\max\left(t_1+d_1,\cdots,t_n+d_n\right)-\max\left(t_1,\cdots,t_n\right)\right)$$

Definition 19 (DMAX). *The distrust maximizing trust score aggregation operator $DMAX$ is defined as*

$$DMAX\left((t_1,d_1),\cdots,(t_n,d_n)\right) = \\ \left(\max\left(t_1+d_1,\cdots,t_n+d_n\right)-\max\left(d_1,\cdots,d_n\right),\max\left(d_1,\cdots,d_n\right)\right)$$

Definition 20 (KAV). *The knowledge preference averaging trust score aggregation operator KAV is defined as*

$$KAV\left((t_1,d_1),\cdots,(t_n,d_n)\right) = \quad (p,q)$$

with (p,q) such that

$$p=\frac{\sum_{i=1}^{n} w_i\cdot t_i}{\sum_{i=1}^{n} w_i},\quad q=\frac{\sum_{i=1}^{n} w_i\cdot d_i}{\sum_{i=1}^{n} w_i} \tag{1}$$

$$w_i=\begin{cases}1 & \text{if } t_i+d_i=\max\left(t_1+d_1,\cdots,t_n+d_n\right)\\ 0 & \text{otherwise}\end{cases}$$

Definition 21 (KMAX). *The knowledge maximizing trust score aggregation operator $KMAX$ is defined as*

$$KMAX\left((t_1, d_1), \cdots, (t_n, d_n)\right) = \left(\max\left(t_1, \cdots, t_n\right), \max\left(d_1, \cdots, d_n\right)\right)$$

Note that this operator corresponds to $\oplus$, the join of the information lattice $([0,1]^2, \leq_k)$ in the trust score space.

Proposition 3. *TMAX, DMAX, KAV and KMAX fulfill Definitions 9-12 and 16. Furthermore, KMAX also satisfies Definitions 13-15 and 17, while TMAX and DMAX also fulfills Definition 15.*

4. Advanced Trust Score Aggregation Operators

Although the four operators from Section 3.2 are perfectly justifiable, it will often be the case that they are too extreme. In some situations, users might prefer also to take into account the opinions from users who have more doubt about their opinions, while in other scenarios they might prefer to listen to less marked opinions instead of only retaining the 'best' and the 'worst' among the opinions.

In this section, we will present two families of aggregation operators which mitigate the behavior of KMAX and KAV, the former by introducing maximum-like weights, and the latter through the incorporation of knowledge defects. Along the way, it will turn out that some theoretical properties need to be sacrificed, or at least adjusted.

4.1. Ordered Weighted Averaging Trust Score Aggregation

A straightforward solution for mitigating KMAX's behavior is to introduce weights in the aggregation process. One of the best-known weighted aggregation strategies is the ordered weighted averaging (OWA) family. The weights can then be chosen in such a way that KMAX's behavior is alleviated, but without harming the underlying maximum thought.

4.1.1. The classical OWA operator

The traditional OWA operator [5] models an aggregation process in which a sequence V of n scalar values are ordered decreasingly and then weighted according to their ordered position by means of a weight vector $W = \langle w_i \rangle$,

such that $w_i \in [0,1]$ and $\Sigma_{i=1}^{n} w_i = 1$. In particular, if c_i represents the i^{th} largest value in V,

$$OWA_W(V) = \sum_{i=1}^{n} w_i c_i \tag{2}$$

The OWA's main strength is its flexibility, since it enables us to model a whole range of aggregation strategies: e.g., if $W_1 = \langle 0, \cdots, 0, 1 \rangle$, then OWA_{W_1} equals min; the average is modeled by $W_2 = \langle 1/n, \cdots, 1/n \rangle$, etc. Moreover, the reordering of the arguments introduces an element of non-linearity into an otherwise linear process.

OWA operators can be analyzed by several measures, among which the orness-degree that computes how similar its behavior is to that of max:

$$orness(W) = \frac{1}{n-1} \sum_{i=1}^{n} ((n-i) \cdot w_i) \tag{3}$$

For example, $orness(W_1) = 0$, while the weight vector $\langle 0.8, 0.1, 0.1, 0, 0 \rangle$ yields an orness-degree of 0.925, meaning that its behavior is very maximum-like. Yager proved that the higher the orness, the more the operator approaches the maximum [17]. OWA operators with an orness degree less than 0.5 approach the minimum operator.

The application of an OWA operator requires scalar values as arguments. As such, OWA operators are not directly applicable to aggregate trust scores. Therefore, we propose to perform trust aggregation by means of two separate OWA operators, one for the trust and one for the distrust degrees.

4.1.2. OWA Operators for Trust Scores

Below, we describe a generic procedure for applying (standard) OWA operators to the trust score aggregation problem:

1. Determine n, the number of trust score arguments distinct from $(0,0)$. Trust scores that represent complete ignorance do not take part in the aggregation process[5].

[5]If all trust scores equal $(0,0)$, the final result is also set to $(0,0)$ and the aggregation process terminates at this step.

2. Construct the sequences T and D, containing the n trust values (resp., the n distrust values) of the trust score arguments.
3. Construct n-dimensional weight vectors W_T and W_D.
4. Compute the aggregated trust score as $(OWA_{W_T}(T), OWA_{W_D}(D))$.

If we add an extra restriction to the construction of W_T and W_D which ensures that the orness degree of both weight vectors is at least 0.5 (so that they exhibit a maximum-like behavior), the above procedure can be generalized into a class of trust score aggregation operators that can be seen as alternate, mitigated, versions of KMAX.

Definition 22 (K-OWA). *We define the trust score OWA operator K-OWA_{W_T,W_D} associated with the trust weight vector W_T and distrust weight vector W_D as*

$$K\text{-}OWA_{W_T,W_D}((t_1,d_1),\cdots,(t_n,d_n)) = \\ (OWA_{W_T}(t_1,\cdots,t_n), OWA_{W_D}(d_1,\cdots,d_n)),$$

with $W_T = \langle w_{T_1},\cdots,w_{T_n}\rangle$ and $W_D = \langle w_{D_1},\cdots,w_{D_n}\rangle$ such that

$$orness(W_T) \geq 0.5 \text{ and } orness(W_D) \geq 0.5$$
$$\sum_{i=1}^{n} w_{T_i} = 1 \text{ and } \sum_{i=1}^{n} w_{D_i} = 1$$

with $i = \{1,\cdots,n\}$ and $(t_i,d_i) \neq (0,0)$.

Proposition 4. *K-OWA always fulfills Definitions 9, 10, 12a, 12b and 16 (regardless of the weight choice). In order for Definition 17 to hold, it suffices that $w_{T_i} = 0$ and $w_{D_i} = 0$ as soon as $i > \frac{n}{2}$.*

This can be verified by construction and by the properties of an OWA operator. Definitions 13 and 14 are harder conditions to fulfill in general, as the following example shows.

Example 5. *If the trust scores to aggregate are $(1,0)$, $(0,0)$, $(0,0)$ and $(0,0)$, then the outcome by our OWA procedure is $(1,0)$ (regardless of the choice of weights vectors, because $n = 1$). If we change these trust scores to $(1,0)$, $(0.1,0)$, $(0.1,0)$ and $(0.1,0)$, the number of arguments that take part in the OWA aggregation equals 4. If we compute the weights for instance as $W_T = \langle\frac{2}{3},\frac{1}{3},0,0\rangle$ and $W_D = \langle 1,0,0,0\rangle$, the final result of the aggregation equals $(0.7,0)$. So, although $(0,0) \leq_{td} (0.1,0)$ and $(0,0) \leq_k (0.1,0)$, $(1,0) \not\leq_{td} (0.7,0)$ and $(1,0) \not\leq_k (0.7,0)$.*

The reason for the failure of the monotonicity properties in this example is due to the presence (and subsequent alteration) of $(0, 0)$ trust score arguments, which causes the application of the OWA operators to a different number of arguments. It can be verified, however, that if we add the restriction to Definitions 13 and 14 that $(t_j, d_j) \neq (0, 0)$ and $(t'_j, d'_j) \neq (0, 0)$, the property holds, regardless of the weight vectors W_T and W_D, provided they remain fixed. Analogously for Definition 15.

Note that the associativity condition will not always be fulfilled, since this depends on the weighting scheme (see Example 6). Because of the averaging nature of K-OWA, also the knowledge boundary condition does not always hold: take e.g. $(1, 0)$, $(0.5, 0)$ and $(0.6, 0)$ and $W_T = \langle 1/3, 1/3, 1/3 \rangle$, which yields $(0.7, 0)$ as aggregated result, and $0.7 + 0 < 1 + 0$. In other words, the result of the aggregation will not always reside in the gray triangle of Figure 6. However, it will remain in the area that is marked out by the minimum and maximum among the trust and distrust degrees (since the trust and distrust boundaries are always satisfied).

4.1.3. Determining the weights

As is clear from the above, the actual way of aggregating the trust scores is determined by the choice of the weight vectors. One strategy is to construct W_T and W_D beforehand. For instance, the final trust (resp., distrust) value can be evaluated as the extent to which a predefined fraction (at least one, all of them, a majority, ...) of the trust score arguments exhibits trust (resp., distrust).

Example 6 (Fixed weights). *In [3], given n trust score arguments to aggregate (all distinct from $(0, 0)$), trust and distrust weights are computed by*

$$w_{T_i} = \frac{2 \cdot \max(0, \lceil \frac{n}{2} \rceil - i + 1)}{\lceil \frac{n}{2} \rceil (\lceil \frac{n}{2} \rceil + 1)} \quad , \quad w_{D_i} = \frac{2 \cdot \max(0, \lceil \frac{n}{4} \rceil - i + 1)}{\lceil \frac{n}{4} \rceil (\lceil \frac{n}{4} \rceil + 1)}$$

The disparity between trust and distrust weights was motivated by the observation that a few distrust statements about x (in particular, a quarter of them) may suffice to reach a final conclusion of distrust, while the evaluation of trust depends on the majority of the arguments; distrust is easier established than trust. Note that weights are decreasing, in a sense that the higher trust/distrust values have a stronger impact than the lower ones.

This example also illustrates why K-OWA does not always satisfy the associativity condition: let $n = 3$, if we aggregate all three trust scores at

once, two of them will take part in the trust degree computation; however, when we perform the aggregation in two phases, only one trust score will take part in each trust degree computation.

The above example can be generalized into an implementation of the K-OWA family, where the weight vectors are determined by two parameters α and β:

Example 7 (Fixed weights family). *Let α and β in $[1, \infty]$. The weights can then be computed as (with $i = \{1, \cdots, n\}$ and $(t_i, d_i) \neq (0, 0)$):*

$$w_{T_i} = \frac{2 \cdot \max(0, \lceil \frac{n}{\alpha} \rceil - i + 1)}{\lceil \frac{n}{\alpha} \rceil (\lceil \frac{n}{\alpha} \rceil + 1)}$$
$$w_{D_i} = \frac{2 \cdot \max(0, \lceil \frac{n}{\beta} \rceil - i + 1)}{\lceil \frac{n}{\beta} \rceil (\lceil \frac{n}{\beta} \rceil + 1)}$$

Remark that KMAX is a special case of this particular implementation of K-OWA, with $\alpha = \beta = n$.

Proposition 5. *The trust score aggregation operators from Example 7 always exhibit a maximum-like behavior, regardless of the α and β choice.*

Proof. We show that the orness degree of W_D and W_T will never be lower than 2/3: the lowest possible orness for W_T and W_D will be achieved when all arguments take part in the aggregation process, i.e., when $\alpha = \beta = 1$. This yields weights $W_{T_i} = W_{D_i} = \frac{2(n-i+1)}{n(n+1)}$. Consequently,

$$orness(W_T) = orness(W_D) =$$
$$\frac{1}{n-1} \sum_{i=1}^{n} ((n-i)(\frac{2(n-i+1)}{n(n+1)})) =$$
$$\frac{2\left(\sum_{i=1}^{n} (n-i)^2 + (n-i)\right)}{n(n-1)(n+1)} =$$
$$\frac{2\left(\sum_{i=1}^{n} (n^2+n) + \sum_{i=1}^{n} (i^2-i) - 2n \sum_{i=1}^{n} i\right)}{n(n-1)(n+1)} =$$
$$\frac{2\left(n^3+n^2+1/3n^3+1/2n^2+1/6n=1/2n^2+1/2n-2/2n^3-2/2n^2\right)}{n(n-1)(n+1)} =$$
$$2/3.$$

4.2. Knowledge-Enhanced Trust Score Aggregation

In Section 3.2 we explained that the KAV trust score aggregation operator exhibits the most moderating behavior among the bilattice-based strategies. However, one can argue that, for some applications/situations, even KAV might be too extreme, since it only takes into account the opinions of the most knowledgeable agents. This is illustrated by the following example.

Example 8. *Assume that the trust scores to aggregate are* $(1, 0)$, $(1, 0)$, $(0.7, 0.29)$ *and* $(0.7, 0.29)$. *In other words, two of the trust score arguments have perfect knowledge (*$kd(1, 0) = 0$*), while the other only show a very small knowledge defect (*$kd(0.7, 0.29) \approx 0$*). Intuitively, one would expect their contribution to be almost equally important, and hence that the final trust degree lies somewhere in the middle between* 0.7 *and* 1*, and the distrust degree between* 0 *and* 0.29*. However, the KAV aggregated trust score equals* $(1, 0)$.

In a way, the determination of the weights for the KAV aggregation can be seen as a binary process, because only the users with the most perfect knowledge (in other words, with the lowest knowledge defect) take part in the aggregation, even if the difference with some of the other arguments is almost negligible. A possible solution to this problem is the following new family of trust score aggregation operators; they mitigate the behavior of KAV, but without harming the underlying thought of trust score discrimination w.r.t. knowledge amount.

Definition 23 (KAAV). *The knowledge awarding averaging trust score aggregation operator* $KAAV_\gamma$ *associated with knowledge reward* $\gamma \in [0, \infty]$ *is defined as*

$$KAAV_\gamma\left((t_1, d_1), \cdots, (t_n, d_n)\right) = \quad (p, q)$$

with (p, q) *such that*

$$p = \sum_{i=1}^{n} w_i \cdot t_i \text{ and } q = \sum_{i=1}^{n} w_i \cdot d_i$$

$$w_i = \frac{(1 - kd(t_i, d_i))^\gamma}{\sum_{i=1}^{n} (1 - kd(t_i, d_i))^\gamma}$$

If all trust scores have $kd(t, d) = 1$*, then the aggregated result is* $(0, 0)$.

If the knowledge reward γ equals 0, then we obtain the arithmetic mean. When $\gamma = 1$, each trust score is weighted inversely proportional to its knowledge defect: the lower $kd(t, d)$, the higher the associated weight. Note that the aggregated trust score will approximate KAV's result for $\gamma \to \infty$.

Example 9 (Knowledge-dependent weights). *In the case of Example 8, with* $\gamma = 1$, $W = \langle \frac{1}{3.98}, \frac{1}{3.98}, \frac{0.99}{3.98}, \frac{0.99}{3.98} \rangle$. *Then the aggregated trust score* $(p, q) \approx (0.85, 0.15)$, *a much more intuitive result.*

Proposition 6. *KAAV fulfills Definitions 9 and 10, 12a, 12b, 16 and 17.*

The following counterexample demonstrates that the associativity condition does not hold in general: $KAAV_1((1,0),(0.5,0),(0.2,0.8)) = (0.58, 0.32) \neq KAAV_1(KAAV_1((1,0),(0.5,0)),(0.2,0.8)) \approx (0.49, 044)$. Note that, as was the case with K-OWA too, mitigating the behavior implies we have to sacrifice the knowledge boundary condition. Note that trust, distrust, trust-distrust and knowledge monotonicity do not hold in general because of the way the weights are generated: even if $t_j > t_{j'}$ and hence $1-kd(t_j, d_j) > 1-kd(t_{j'}, d_{j'})$ since we assume consistent agents, this does not imply that the final aggregated trust score p will be higher than $p^{'}$, since the other weights are also affected by the change in the jth weight. E.g., $(0.5, 0.3) \geq_d (0, 0.3)$, but $A((1,0),(1,0),(0.5,0.3)) \approx (0.870, 0.039) <_d A((1,0),(1,0),(0,0.3)) \approx (0.857, 0.086)$. As a result, trust-distrust and knowledge monotonicity do not hold.

5. Aggregation Operators in Practice

While the previous sections were concerned with the rationale and theory behind the aggregation operators, in this section, we investigate their usefulness in practice. For extra analyses and comparisons between the aggregation operators, we refer to [18].

5.1. Data Set and Methodology

The data set we use in our experiments is obtained from CouchSurfing, a web application with a strong social networking component. After a couch experience, users can fill in a form to evaluate their guest or host. One of the questions assesses the trust relationship between the parties involved: users can indicate whether they 'don't trust', 'somewhat', 'generally' or 'highly' trust another user, trust him/her 'with his life', or have no idea ('don't know'). The

Table 1: Distribution of trust and knowledge statements in the data set

trust statements				*knowledge statements*			
type	t'	#	%	*type*	k	#	%
don't know	0	152 443	5.65	not at all	0	88 808	3.29
don't trust	0	6 429	0.24	a little bit	0.25	573 517	21.26
somewhat trust	0.25	331 061	12.27	somewhat	0.5	761 085	28.21
generally trust	0.5	994 476	36.86	fairly well	0.75	658 996	24.43
highly trust	0.75	893 428	33.12	very well	1	341 529	12.66
trust with my life	1	319 868	11.86	extremely well	1	177 001	6.56
				couldn't know any better	1	96 770	3.59

trust field is kept private to other members and is not mandatory. Nonetheless, these relationships constitute a large trust network of 397 471 users and 2 697 706 trust statements. The distribution of the network is given in Table 1. Furthermore, the users can also indicate how well they know each other, using one of the following options: 'not at all', 'a little bit', 'somewhat', 'fairly well', 'very well', 'extremely well' or 'could not know any better'.

This data set is very suitable for our purposes, since it contains gradual trust, distrust, and knowledge levels. Unfortunately, the CouchSurfing data does not perfectly align with our trust score space setting (e.g., the latter requires cardinal values, whereas the data set only provides ordinal ones); hence, we need a heuristic method to map the available information into trust scores. Our translation of the trust and knowledge statements into $[0, 1]$ values can be found in Table 1. We translate the three highest knowledge levels to the maximum knowledge value 1 due to pragmatic reasons, to ensure a more balanced distribution of the trust scores over the trust score space.

We propose to map the available trust and knowledge information to trust scores according to the following formula: $(t, d) = \big(k \cdot t', k \cdot (1 - t')\big)$, with t' (k) the translation of the trust (knowledge) statement. In this way, we obtain consistent trust scores with the desirable properties $t + d = k$ and $1 - k = kd(t, d)$, and a data set with a high variety of knowledge defects and gradual trust and distrust degrees. In our experiments, we ignore the records that contain a 'not at all' knowledge statement or 'don't know' trust level, since we interpret $(0, 0)$ as no link. We do retain $(0, 0)$ trust scores which are the result of propagation.

To measure the performance of the aggregation operators, we use the leave-

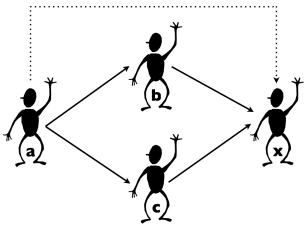

Figure 7: Example of a leave-one-out scenario

one-out method, which consists of hiding a trust relation and trying to predict its hidden value (i.e., the trust score). Figure 7 depicts an example of a trust network which contains four users with five trust relations: we hide the trust relation from user a to user x (dotted line) and try to estimate its trust score by aggregating the opinions of a's acquaintances b and c. Therefore, we need to propagate a's trust score in b and b's trust score in x, and a's trust score in c and c's trust score in x. In our experiments, we take into account propagation paths of exactly length 2 and use the P_1 propagation operator of Definition 7 (because it is at the basis of all other propagation operators) with $\mathcal{T} = \min$. In this way, we can ensure that every aggregate is the result of the same propagation process and that there is no variable path length which might introduce extra errors, making it harder to study the actual impact of the aggregation operators. In this configuration, the CouchSurfing data set consists of 1 596 674 distinct a-x trust scores with at least one a-b-x trust path between them.

To measure the accuracy of the aggregation operators, we work with two measures that are variations on the classical mean absolute error (MAE) and root mean squared error (RMSE). The first measure considers every error of equal value, while the latter one emphasizes larger errors. The computation of the MAE and RMSE is done by determining the deviation between the hidden trust score and the predicted trust score for each leave-one-out experiment. Since MAE and RMSE are defined for scalar values, but we work with trust scores which have two components, we define the trust score MAE as the average of the Manhattan distances and the trust score RMSE as an adjusted form of the Euclidean distances:

Definition 24 (T-MAE, T-RMSE). *We define the trust score MAE and*

RMSE in a leave one out experiment with n trust score predictions as

$$\begin{aligned} \textit{T-MAE} &= \frac{\sum_{i=1}^{n} |t_{r_i} - t_{p_i}| + |d_{r_i} - d_{p_i}|}{n} \\ \textit{T-RMSE} &= \sqrt{\left(\sum_{i=1}^{n} (t_{r_i} - t_{p_i})^2 + (d_{r_i} - d_{p_i})^2 \right) / n}, \end{aligned}$$

with t_{r_i} (d_{r_i}) the real trust (distrust) degree and t_{p_i} (d_{p_i}) the predicted trust (distrust) degree.

Since the trust and distrust degrees range between 0 and 1, the extreme values that T-MAE can reach are 0 and 2, while T-RMSE $\in [0, \sqrt{2}]$.

5.2. Comparative Analysis of the Bilattice-Based Operators

In this section, we will discuss the performance of the aggregation operators introduced in Section 3.2, and compare them with two additional baseline strategies (the squares E and F in Figure 6): KMIN (E) computes the aggregated trust score as $(t, d) = (\min(t_1, \cdots, t_n), \min(d_1, \cdots, d_n))$, and 'Fixed values' (F) is a strategy that always yields $(0.431, 0.206)$, which represents the average trust and distrust degree in the translated data set. Remark that, unlike TMAX, DMAX, KMAX and KAV, the results of the last two operators do not always reside in the gray area of Figure 6. Table 2 contains the results.

The trust score aggregation operators (upper part of the table) perform more or less the same when considering T-RMSE; however, an inspection of the T-MAE values shows that TMAX and KAV achieve slightly better results than DMAX and KMAX. The baselines (lower part of the table) clearly produce the highest MAE errors, but the RMSE's are the opposite way round for the fixed values baseline, which means that it makes less large prediction errors than the others. This is due to its better prediction for leave-one-out experiments that only contain one path (FIX yields the average trust and distrust degree in the data set, whereas the other algorithms just copy the input). At first glance, it looks as if there is no clear winner among TMAX, DMAX, KMAX and KAV. However, in the remainder of the discussion, we will show that overall T-MAE's and T-RMSE's do not give a complete picture of an

Table 2: Overall performance of aggregation strategies on the CouchSurfing data set, with propagation operator P_1 and $\mathcal{T} = \min$ for paths of length 2; T-MAE $\in [0, 2]$ and T-RMSE $\in [0, \sqrt{2}]$.

Aggregation operator		Figure 6	*T-MAE*	*T-RMSE*
Trust maximizing	TMAX	A	0.316	0.321
Distrust maximizing	DMAX	B	0.325	0.324
Knowledge preference averaging	KAV	C	0.318	0.321
Knowledge maximizing	KMAX	D	0.322	0.324
Knowledge minimizing	KMIN	E	0.389	0.389
Fixed values	FIX	F	0.340	0.311

operator's performance, and that sacrificing the knowledge condition allows for other operators that can produce more accurate trust estimations.

Figure 8 shows the evolution of T-MAE[6] over the number n of trust scores that need to be aggregated. The split-up of the results gives us a clearer image than overall MAE errors. Notice that all our non-baselines operators perform more or less equally for small n, but that these classes are exactly the ones that are overrepresented in our experiment. The latter is depicted by the bars, their scale can be found on the right side of the graph. The bars illustrate for example that in more than 500 000 leave-one-out experiments there was only one propagation path and in almost 300 000 cases exactly two (the scenario of Figure 7), as opposed to about 1000 leave-one-out experiments which have to aggregate between 50 and 75 trust scores. These numbers explain why we get very similar average errors in Table 2.

On average, one can see that it becomes more difficult to produce accurate predictions as the number of inputs starts to increase, and that there is clearly a performance difference between the operators: DMAX and KMAX make a very bad showing from the moment they have to deal with more than

[6]Similar results are obtained for T-RMSE.

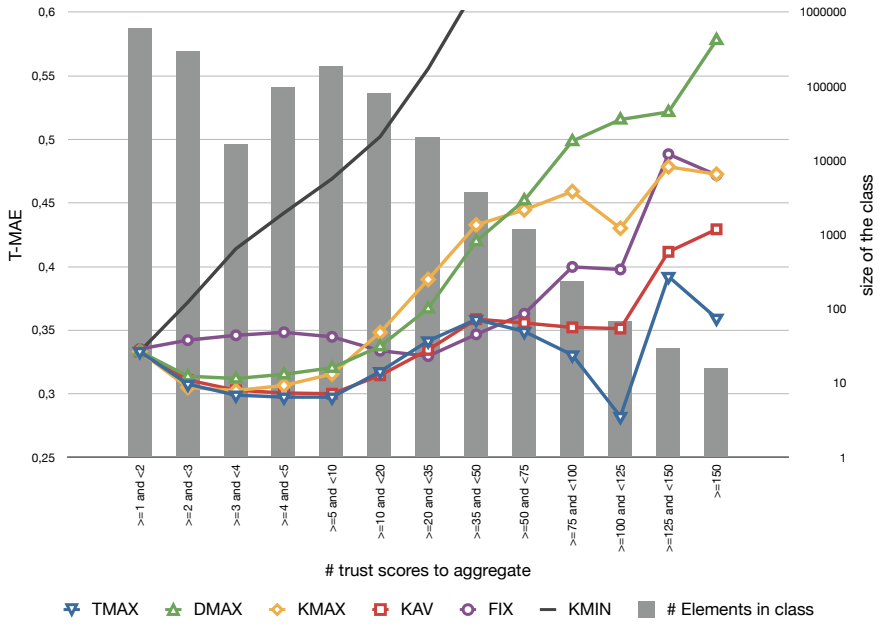

Figure 8: T-MAE for the trust score aggregation operators; split-up according to the number of aggregates.

20 inputs; their T-MAE is often almost twice as high as TMAX's. Obviously, these two operators are too extreme for the CouchSurfing application. TMAX and KAV adapts themselves much better to changing aggregation conditions. Note that TMAX achieves somewhat lower errors in cases with more than 75 inputs, which can be explained by the fact that the average trust degree t_{r_i} for $n > 75$ is significantly higher than for $n \leq 75$ (viz. 0.594 vs. 0.423).

5.3. The Effect of Orness Weights for KMAX

Table 2 and Figure 8 demonstrated that KMAX's attitude is too explicit. As we discussed in Section 4.1.2, a possible solution is the introduction of or-like weights. We will illustrate the effect of tuning the weight vectors for K-OWA and show that they can indeed have a positive impact on the T-MAE. The goal of this experiment is to illustrate the importance of choosing the right trust and distrust weights, and not to compute the ideal weighting strategy (the latter can for instance be achieved by automatic methods such as machine learning techniques). To this aim, we use the proposals in Example 7.

In Figure 9, we compare KMAX's performance with some of the operators of the K-OWA family. KMAX is represented by the circles, while the mitigation on the trust and distrust side (tuning of α and β resp.) is represented by the triangles and the inverse triangles respectively. The line with the squares depicts the course of a K-OWA operator with $\alpha \neq n \neq \beta$.

Only introducing weights on the trust side does not have the effect we wished for: there is a small positive impact for aggregation conditions with less than 50 inputs, but for higher numbers the situation is perceptibly deteriorating. Note that the results are worse the further away from the maximum behavior: taking into account the first three quarters of the ordered arguments ($\alpha = 4/3$) yields worse results than only taking into account a quarter ($\alpha = 4$), which is worse than only taking into account the first ordered argument (maximum, KMAX, $\alpha = n$). The more sources to aggregate, the larger the possible effect of orderings and weight vectors; the effect of maximizing knowledge becomes more important when we have to deal with many inputs. This finding is also confirmed by the good results of TMAX (which maximizes the trust degree) for aggregation conditions with a high number of inputs.

It is clear that changing the orness-weights on the distrust side has an overall positive effect, and that better results are achieved when taking into account opinions from many sources, for conditions for both low and high n values. Recall that DMAX (maximizing the distrust degree) performed very bad for high n (see Figure 8); the inverse triangle results in Figure 9 show the benefits of making DMAX's conduct less explicit.

Obviously, the optimal weighting scheme for K-OWA lies somewhere in between the extremes TMAX and DMAX. This is illustrated by the squares, which embody the behavior of K-OWA with $\alpha = 4$ and $\beta = 4/3$, i.e., using the first quarter of the highest trust estimations and three quarters of the highest distrust estimations. This means that, in the context of CouchSurfing, trust is easier established than distrust; it is an open, voluntary, community of users who want and have to rely on each other. Note that the benefit of using K-OWA to determine the trust and distrust level is especially high for intermediate aggregation conditions (> 10 and ≤ 50), with T-MAE decreases of 25% compared to KMAX.

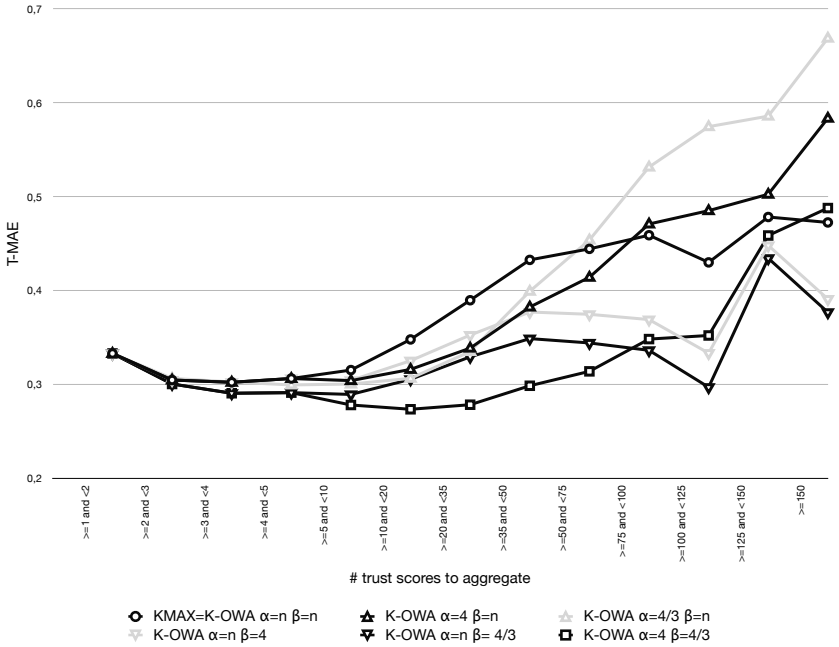

Figure 9: Tuning the weights for K-OWA; $\alpha = \beta = n$ in Example 7 yields KMAX.

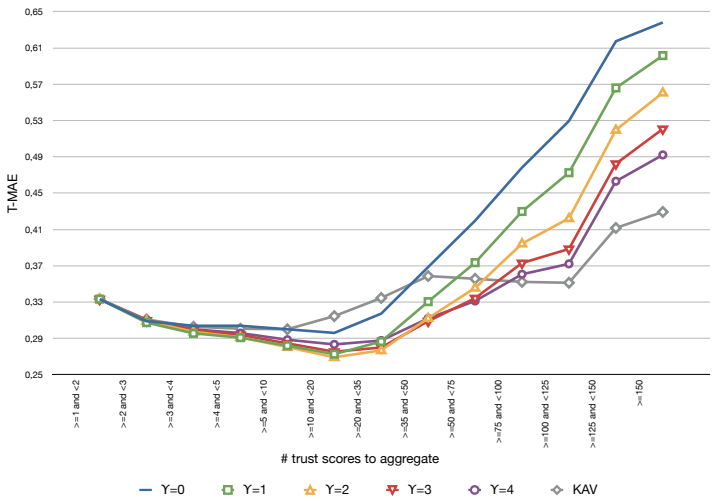

Figure 10: Tuning the knowledge reward γ for KAAV; $\gamma = 0$ results in the arithmetic mean, and $\gamma \to \infty$ yields KAV.

5.4. The Effect of Knowledge Incorporation for KAV

The knowledge preference averaging operator KAV only takes into account the opinions of the most knowledgeable users. In Section 4.2 we claimed that its performance might be improved if we incorporate additional opinions, by rewarding those that are more certain (the KAAV operator). In this section, we show that this is indeed the case. Again, the goal of the experiment is not to compute the most optimal implementation of KAAV, but to illustrate its advantages over the KAV operator.

In Figure 10 we compare the results achieved for the arithmetic mean (all opinions are equally important, KAAV with $\gamma = 0$, solid line), three implementations of KAAV with increasing knowledge reward γ, and KAV (only the most knowledgeable opinions matter, diamonds). Note that for a very large γ, KAAV reduces to KAV. In other words, the line that is determined by the knowledge defect of the trust score resulting from the KAAV aggregation will approximate the corresponding KAV line for increasing knowledge awards.

First note that this graph again illustrates that it becomes more difficult to accurately predict trust scores when a lot of opinions are available. Let us now compare the classical average with the knowledge preference average KAV. For only few inputs, the average performs better than KAV. However, for larger numbers n of aggregates, the average fails completely, with T-MAE increases up to 50% compared to KAV. This demonstrates that knowledge-enhanced aggregation strategies become more useful when many inputs have to be aggregated. The trend is also apparent when focusing on the KAAV implementations, with $\gamma = 1, \ldots, 4$: as the number of inputs increases, KAAV's with larger knowledge awards achieve better results and come closer to KAV's trust score MAE. On the other hand, remark that mitigating KAV's behavior produces better results for smaller n's, even lower than those of the arithmetic mean.

The results show that, when choosing between the average, KAV and KAAV, a trade-off should be made between lower errors for smaller n and lower T-MAE's for aggregation conditions with many inputs. The best configuration for a particular application can then e.g. be estimated during a training phase.

5.5. Discussion

Although from a theoretical perspective, not satisfying the knowledge boundary or monotonicity properties can be regarded as serious drawbacks for any trust score aggregation operator, our experiments demonstrate that dropping these conditions opens the door for new, more practical, aggregation operators. Our results on a large real-world data set from CouchSurfing showed that this kind of operators can significantly improve the performance of the less sophisticated, theoretically sounder, aggregation operators.

These findings can be explained by the imperfect, noisy nature of our data, a problem inherent in the larger part of social network applications. Some users might not fully understand the meaning of each trust/knowledge statement, others make an unintentional mistake (e.g. checked the wrong trust box, forgot to indicate the knowledge level, ..), and so on. Furthermore, the data in trust-enhanced applications must be mapped to a practical model, and the trust estimation mechanisms must be captured as precisely as possible. However, the resulting trust model and propagation operators remain approximations, and hence will always introduce some extra noise.

One way of reducing the noise in the data is by making the aggregation operators less extreme, e.g. by using knowledge-enhanced strategies that give priority to better informed opinions, or by introducing weights that soften the standard behavior of the operators. This explains the good performance of the KAAV and K-OWA families, compared to the classical operators.

Moreover, one might also argue that the bilattice-based operators perform less because the properties we enforced do not align well with the CouchSurfing data. It turns out that the knowledge boundary condition is not as vital as we thought, compared to the trust and distrust boundaries, the neutrality and opposite arguments condition. The former property is exactly the one that is only fulfilled for the bilattice-based approaches, whereas the KAAV and K-OWA families do satisfy the latter.

The choice as to which approach (TMAX, KAV, KAAV, K-OWA, ...) is most suitable also depends on the application at hand. In applications where prudence is in order, one can e.g. opt for a K-OWA operator with a large β-parameter (which results in a higher orness degree and hence will sooner yield a high aggregated distrust degree). Or, in large user networks where (partial) ignorance is the rule rather than the exception, KMAX or KAAV might be preferred over TMAX or K-OWA.

6. Conclusions and Future Work

Research in trust networks is still in its infancy, in particular when it comes down to the representation, propagation and aggregation of distrust. In this paper, we have built upon previous work in which trust scores are modeled as (trust,distrust)-couples that are drawn from a bilattice, and investigated which requirements a trust score aggregator needs to fulfill. Based on these aggregation conditions, we proposed four trust score aggregation strategies, each with their own distinct behavior: the trust maximizing operator TMAX which is the most optimistic choice (maximum trust degree for the lowest possible knowledge level), distrust maximizing DMAX which is the most pessimistic one (maximum distrust degree), knowledge preference averaging KAV which is the most moderating approach (average of the most knowledgeable trust scores), and the knowledge maximizing operator KMAX (maximum trust and distrust degree), the boldest aggregation option.

Besides, we also introduced two other families of operators: the OWA-based K-OWA operator and the knowledge awarding averaging trust score operator KAAV, mitigating the behavior of KMAX and KAV respectively. Although K-OWA and KAAV have less desirable properties from a theoretical perspective, our experiments on a large data set from CouchSurfing demonstrated that they achieve more accurate results in real-world social applications, which are inherently noisy.

Obviously, the reported performances do not only depend on the choice of aggregation operator, but also on the combination with propagation, which inherently introduces errors in the computation too. Hence, a first step in our future research is the investigation of the synergy between the two operator types and their separate influence on the accuracy.

A second research direction involves the further exploration of the role of knowledge defects; we embarked on this topic in [4]. For instance, it makes sense to generate weights which not only depend on the orness-degree, but also on the amount of knowledge that the corresponding trust scores contain (resulting in a variation of K-OWA with knowledge-dependent weights). Another way to include knowledge information is by using induced OWA operators [19] which order the elements based on increasing knowledge defects.

Thirdly, we also aim to refine the aggregation strategies to take into account certain aspects of the virtual trust network's topology. In particular, the current approaches are indifferent as to the length of the paths that

generated the individual trust scores, and also do not consider how many times the same user appears on a path; see [20] for initial results.

As another part of our future work, we also want to investigate the suitability of our operators for specific trust-enhanced applications. In particular, we are focusing on recommender systems, applications which suggest items to users who might be interested in them [21, 22, 23, 24]. Trust information can help to establish more, and more accurate, recommendations, and its incorporation into existing recommender system technology is a topic of ongoing research (see e.g. [10, 25, 26]).

Finally, another interesting research path is the investigation of the applicability of concepts from the preference modeling area (transitivity, incomplete preference) in the context of trust score aggregation: preference relations that do not necessarily satisfy reciprocity have been studied in [27]; trust scores can be identified with a couple of such preference evaluations, which may provide us more insight into the management and processing of trust and distrust degrees.

Acknowledgments

The authors thank CouchSurfing for making the data available. Patricia Victor would like to thank the Institute for the Promotion of Innovation through Science and Technology in Flanders for funding her research, and Chris Cornelis the Research Foundation–Flanders for funding his 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. Victor, C. Cornelis, M. De Cock, P. Pinheiro da Silva, Gradual trust and distrust in recommender systems, Fuzzy Sets and Systems 160 (2009) 1367–1382.

[2] M. Ginsberg, Multi-valued logics: A uniform approach to reasoning in artificial intelligence, Computational Intelligence 4 (1988) 265–316.

[3] P. Victor, C. Cornelis, M. De Cock, E. Herrera-Viedma, Aggregation of gradual trust and distrust, in: Proceedings of Eurofuse Workshop on Preference Modeling and Decision Analysis (Eurofuse 2009), 2009, pp. 259–264.

[4] C. Cornelis, P. Victor, E. Herrera-Viedma, Ordered weighted averaging approaches for aggregating gradual trust and distrust, in: Proceedings of ESTYLF 2010, 2010, pp. 555-560.

[5] R. Yager, On ordered weighted averaging aggregation operators in multicriteria decision making, IEEE Transactions on Systems, Man, and Cybernetics 18 (1988) 183–190.

[6] M. De Cock, P. Pinheiro da Silva, A many-valued representation and propagation of trust and distrust, Lecture Notes in Computer Science 3849 (2006) 108–113.

[7] H. Prade, A qualitative bipolar argumentative view of trust, Lecture Notes in Artificial Intelligence 4772 (2007) 268–276.

[8] A. Abdul-Rahman, S. Hailes, Supporting trust in virtual communities, in: Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000, pp. 1769–1777.

[9] F. Almenárez, A. Marín, C. Campo, C. García, PTM: A pervasive trust management model for dynamic open environments, in: Proceedings of the First Workshop on Pervasive Security, Privacy and Trust, in conjuntion with Mobiquitous, 2004.

[10] J. Golbeck, B. Parsia, J. Hendler, Trust networks on the semantic web, Lecture Notes in Artificial Intelligence 2782 (2003) 238–249.

[11] S. Kamvar, M. Schlosser, H. Garcia-Molina, The eigentrust algorithm for reputation management in p2p networks, in: Proceedings of WWW2003, 2003, pp. 640–651.

[12] L. Mui, M. Mohtashemi, A. Halberstadt, A computational model of trust and reputation, in: Proceedings of the 35th Hawaii International Conference on System Sciences.

[13] S. Noh, Calculating trust using aggregation rules in social networks, Lecture Notes in Computer Science 4610 (2007) 361–371.

[14] M. Richardson, R. Agrawal, P. Domingos, Trust management for the semantic web, in: Proceedings of the ISWC, 2003, pp. 351–368.

[15] R. Guha, R. Kumar, P. Raghavan, A. Tomkins, Propagation of trust and distrust, in: Proceedings of WWW04, 2004, pp. 403–412.

[16] A. Jøsang, S. Marsh, S. Pope, Exploring different types of trust propagation, LNCS 3986 (2006) 179–192.

[17] R. Yager, Families of OWA operators, Fuzzy Sets and Systems 59 (1993) 125–148.

[18] P. Victor, Trust networks for recommender systems, Ph.D. dissertation (2010).

[19] R. Yager, D. Filev, Induced ordered weighted averaging operators, IEEE Transactions on Systems, Man, and Cybernetics 29 (1999) 141–150.

[20] N. Verbiest, C. Cornelis, P. Victor, E. Herrera-Viedma, Strategies for incorporating knowledge defects and path length in trust aggregation, in: Proceedings of IEA-AIE2010, 2010, pp. 450-459.

[21] P. Resnick, H. Varian, Recommender systems, Communications of the ACM 40 (1997) 56–58.

[22] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Tr Knowl Dat En (2005) 734–749.

[23] R. Yager, Fuzzy logic methods in recommender systems, Fuzzy Sets and Systems 136 (2003) 133–149.

[24] C. Porcel, E. Herrera-Viedma, Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries, Knowledge-Based Systems 23 (2010) 32–39.

[25] P. Victor, C. Cornelis, M. De Cock, A. Teredesai, Trust- and distrust-based recommendations for controversial reviews, IEEE Intelligent Systems, in press.

[26] P. Massa, A. Avesani, Trust-aware recommender systems, in: Proceedings of the ACM Recommender Systems Conference (RECSYS 2007), 2007, pp. 17–24.

[27] E. Herrera-Viedma, S. Alonso, F. Chiclana, F. Herrera, A consensus model for group decision making with incomplete fuzzy preference relations, IEEE Transactions on Fuzzy Systems 15 (2007) 863–877.