

# Trust and Distrust Aggregation Enhanced with Path Length Incorporation

Nele Verbiest<sup>1</sup>, Chris Cornelis<sup>1,2</sup>, Patricia Victor<sup>1</sup>, Enrique Herrera-Viedma<sup>2</sup>

<sup>1</sup>*Department of Applied Mathematics and Computer Science, Ghent University, Ghent, Belgium*

<sup>2</sup>*Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain*

---

## Abstract

Trust networks are social networks in which users can assign trust scores to each other. In order to estimate these scores for agents that are indirectly connected through the network, a range of trust score aggregators has been proposed. Currently, none of them takes into account the length of the paths that connect users; however, this appears to be a critical factor since longer paths generally contain less reliable information. In this paper, we introduce and evaluate several path length incorporating aggregation strategies in order to strike the right balance between generating more predictions on the one hand and maintaining a high prediction accuracy on the other hand.

*Key words:* trust networks, aggregation operators, path length

---

## 1. Introduction

Trust is an important factor in many online social networks. For example, consumer websites like Epinions<sup>1</sup> include a recommendation feature where product reviews written by reliable (trusted) peers are suggested to users. The effectiveness of these recommendations relies on the principle that users are more interested in information from people they know, and that users that know each other often have common interests [11]. As another example, in CouchSurfing<sup>2</sup>, which allows users to find or offer free hosting, members can explicitly state their trust or distrust in each other by completing an online form.

Trust networks, designed to model trust relationships between users (called agents in this context), have been studied extensively in scientific literature (see e.g. [1, 2, 5, 9, 12, 13, 16, 17, 19, 20]). In this paper, we model computational trust using the bilattice-based approach of Victor et al. [15], which considers both gradual trust and distrust.

As trust networks may be huge, most agents in the network do not know each other. In [16], four kinds of problems related with computational trust are considered:

1. Estimate a trust score between two agents  $x$  and  $y$ , i.e., determine the level to which  $x$  may trust  $y$
2. Find all agents that are most trusted, or distrusted by a given agent

---

<sup>1</sup> [www.epinions.com](http://www.epinions.com)

<sup>2</sup> [www.CouchSurfing.org](http://www.CouchSurfing.org)

3. Establish as many trust scores as possible between pairs of agents in a trust network
4. Find a measure of trustworthiness of an agent  $x$  based on the opinions of other agents about  $x$

In our work, we focus on the first problem. Various algorithms have been proposed to solve this problem including, among others, TidalTrust [2], Sunny [8] and CloseLook [17]. In the case where trust and distrust are considered simultaneously, seminal contributions were made by Guha et al. [4] and Jøsang et al. [6, 7], who worked within a gradual, respectively a probabilistic trust model. More recently, within the bilattice-based approach Victor [13] set up a comprehensive framework that computes trust-distrust estimations for agent pairs in the network using trust metrics [23]: given two agents in the trust network, we can search for a path between them and propagate the trust scores along this path to obtain an estimation. When more than one path is available, we may single out the most relevant ones (selection), and aggregation operators can then be used to combine the propagated trust scores into one final trust score.<sup>3</sup> In particular, [15] introduced four families of trust score propagation operators, while in [14], a set of reasonable properties for trust score aggregation was introduced, along with a number of aggregation operators that satisfy them.

In this paper, we take this study one step further, by considering also the length of the paths that connect two agents; in general, as the path gets longer, it may be considered less trustworthy since more propagation steps are needed to obtain an estimation. This idea of “trust decay” is often implemented in trust metrics, e.g. by incorporating a spreading factor [23], taking into account only shortest paths [3], or paths whose length does not exceed a given horizon [10]. On the other hand, dismissing longer paths altogether may negatively impact the coverage of the trust metrics, i.e., the fraction of agent pairs for which an estimation can be obtained. Therefore, in this paper we develop two kinds of mechanisms to optimize the trade-off between coverage and accuracy; first, we propose various weighted aggregation strategies that incorporate path length, and secondly we modify the trust estimation algorithm to include a “(semi)-dynamic” horizon, such that longer paths are only considered for those agent pairs which are not connected by a shorter path. We test the usefulness of these strategies on the CouchSurfing network.

The remainder of this paper is structured as follows. In Section 2 we introduce necessary concepts on trust networks, while in Section 3 we introduce our path length incorporating trust aggregation strategies. The experimental evaluation is described in Section 4. Finally, we conclude in Section 5.

## 2. Trust Networks

Trust networks are social networks in which agents can express their opinion about other agents. Formally, a trust network is a pair  $(A, R)$ , with  $A$  a set of agents and  $R$  a function  $A \times A \rightarrow T$ . For a pair of agents  $x$  and  $y$ ,  $R(x, y)$  is called the trust score of  $x$  in  $y$ . The particular structure of the elements in  $T$  refers to the nature of the opinion that the agents can have about each other.

---

<sup>3</sup> Note that in [16], the propagation operation is called composition.

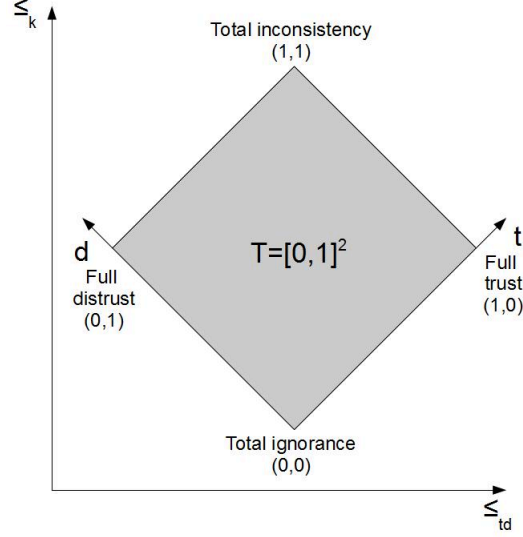


Figure 1: Bilattice  $\mathcal{BL}^\square$  for trust scores.

### 2.1. Bilattice-Based Approach

In this paper, we follow the bilattice approach from [15], which defines trust scores as pairs  $(t, d) \in [0, 1]^2 = T$ , where  $t$  and  $d$  represent the degree of trust and distrust respectively. This approach considers trust and distrust as separate and gradual concepts, and can therefore adequately model the amount of knowledge that is available in expressing a trust opinion. The trust score space is embedded in a bilattice

$$\mathcal{BL}^\square = (T, \leq_{td}, \leq_k, \neg),$$

where

$$\begin{aligned} &\forall (t_1, d_1), (t_2, d_2) \in [0, 1]^2, \\ &(t_1, d_1) \leq_{td} (t_2, d_2) \Leftrightarrow t_1 \leq t_2 \text{ and } d_1 \geq d_2 \\ &(t_1, d_1) \leq_k (t_2, d_2) \Leftrightarrow t_1 \leq t_2 \text{ and } d_1 \leq d_2 \\ &\neg(t_1, d_1) = (d_1, t_1) \end{aligned}$$

The bilattice  $\mathcal{BL}^\square$  is illustrated in Figure 1. The trust scores are ordered from  $(0, 1)$  (full distrust) to  $(1, 0)$  (full trust) by the trust ordering  $\leq_{td}$ . The knowledge ordering  $\leq_k$  orders the trust scores from  $(0, 0)$  (total ignorance) to  $(1, 1)$  (total inconsistency). We define the knowledge degree of a trust score  $(t, d)$  as  $t + d$ . Trust scores  $(t, d) \in T$  for which  $t + d < 1$  holds lack information and are called *incomplete*, while trust scores  $(t, d) \in T$  for which  $t + d > 1$  holds contain conflicting information and are therefore called *inconsistent*. For more details, we refer to [15].

## 2.2. Trust Metrics

Assume  $x$  and  $y$  are two agents which are connected by the trust network, but for which there is no direct link. In this case, we estimate the trust score of  $x$  in  $y$  by using the trust metrics *propagation*, *selection* and *aggregation*. That is, we consider trust scores on paths between agent  $x$  and  $y$  and combine them to obtain an estimation.

Consider a sequence of agents  $z_1, \dots, z_{m-1}$  between  $x$  and  $y$ , such that  $x$  assigns trust score  $(t_1, d_1)$  to  $z_1$ ,  $z_i$  assigns a trust score  $(t_{i+1}, d_{i+1})$  to  $z_{i+1}$  for all  $i \in 1, \dots, m-2$  and  $z_{m-1}$  assigns trust score  $(t_m, d_m)$  to  $y$ . A propagator  $Prop^m : ([0, 1]^2)^m \rightarrow [0, 1]^2$  combines these trust scores to  $Prop^m((t_1, d_1), \dots, (t_m, d_m)) = (t, d)$ , an estimation of the extent to which  $x$  trusts  $y$ , based on a single path connecting them.

Typically, there are several paths connecting agents  $x$  and  $y$ , all of which generate a propagated trust score, we represent this set of paths by  $P_{x,y}$ . In practice, only a subset of  $P_{x,y}$  is taken into account, for reasons of computational complexity and/or to eliminate less informative paths. This step is performed by a selection operator  $Sel$ , which maps a couple  $(x, y)$  of agents to a subset of  $P_{x,y}$ .

Suppose  $n$  paths between  $x$  and  $y$  are considered; propagation along these  $n$  paths results in  $n$  trust scores  $(t_1, d_1), \dots, (t_n, d_n)$ , which can be combined using an aggregator  $Agg^n : ([0, 1]^2)^n \rightarrow [0, 1]^2$ , which yields a final estimation of the trust score of  $x$  in  $y$ .

### 2.2.1. Trust Propagation

Various propagation operators for trust scores in  $\mathcal{BL}^\square$  are discussed in [15]. In this paper, we define the propagation operation by, for  $(t_1, d_1), (t_2, d_2) \in [0, 1]^2$ :

$$Prop((t_1, d_1), (t_2, d_2)) = (\max(\min(t_1, t_2), \min(d_1, d_2)), \max(\min(t_1, d_2), \min(d_1, t_2)))$$

Our choice is motivated by the fact that this operator, which actively incorporates distrust, showed the best performance in the experiments in [13].

Since  $Prop$  is not associative, we have to fix a propagation order when we need to establish an opinion on  $y$  using more intermediate third parties. Here, we assume that a right-to-left evaluation order (backward propagation) is used, i.e., we recursively define, for  $m > 2$ ,

$$Prop^m((t_1, d_1), \dots, (t_m, d_m)) = Prop^2((t_1, d_1), Prop^{m-1}((t_2, d_2), \dots, (t_m, d_m)))$$

### 2.2.2. Trust Selection

A common way to reduce the number of paths used for generating a trust estimation is to impose an upper limit  $h$  on their length, called horizon [9]. For instance, in the experiments in [14],  $h = 2$  was used, meaning that only trust scores resulting from one-step propagation were considered, and consequently all paths have the same length.

A selection strategy which applies specifically to the bilattice-based trust model is the elimination of paths whose propagated trust score is  $(0, 0)$ . Note that since  $(0, 0)$  represents total ignorance, these trust scores can be safely removed beforehand.

### 2.2.3. Trust Aggregation

We recall the following basic trust aggregators from [14]: for  $(t_1, d_1), \dots, (t_n, d_n) \in [0, 1]^2$ ,

$$\begin{aligned} K-MAX((t_1, d_1), \dots, (t_n, d_n)) &= (\max(t_1, \dots, t_n), \max(d_1, \dots, d_n)) \\ T-MAX((t_1, d_1), \dots, (t_n, d_n)) &= (\max(t_1, \dots, t_n), \max(t_1 + d_1, \dots, t_n + d_n) - \max(t_1, \dots, t_n)) \\ D-MAX((t_1, d_1), \dots, (t_n, d_n)) &= (\max(t_1 + d_1, \dots, t_n + d_n) - \max(d_1, \dots, d_n), \max(d_1, \dots, d_n)) \end{aligned}$$

The definition of these aggregators is motivated by a set of boundary conditions, which state that the final trust (resp., distrust) degree should be bounded by the minimum and the maximum of all trust (resp., distrust) degrees, and that the resulting trust score should not contain less knowledge than the maximum knowledge degree of all trust scores. However, since these basic trust aggregators are min/max-based, they are not very flexible. In order to mitigate their behaviour, they can be softened using ordered weighted average (OWA) operators so as to take into account not just the border trust scores.<sup>4</sup> Therefore, in [14], the following OWA-based<sup>5</sup> variation on K-MAX was proposed, for  $(t_1, d_1), \dots, (t_n, d_n) \in [0, 1]^2 \setminus \{(0, 0)\}$ ,  $m_t, m_d \in [1, n]$ :

$$K-OWA_{m_t, m_d}((t_1, d_1), \dots, (t_n, d_n)) = (OWA_{W_{m_t}}(t_1, \dots, t_n), OWA_{W_{m_d}}(d_1, \dots, d_n))$$

where the decreasing weight vector  $W_m = \langle w_1, \dots, w_n \rangle$  is defined by  $w_i = \frac{\max(0, \lceil \frac{n}{m} \rceil - i + 1)}{\frac{\lceil \frac{n}{m} \rceil (\lceil \frac{n}{m} \rceil + 1)}{2}}$ .

It can easily be verified that when  $m_t = m_d = n$ ,  $K-OWA_{m_t, m_d} = K-MAX$ , i.e., only the maximal trust and distrust degree among the participating trust scores are taken into account. When  $m_t$  and  $m_n$  are smaller than 1, the strict behaviour of the maximum operator is softened, allowing other trust scores to influence the result as well, albeit to a smaller extent. In this paper, we consider  $m_t = m_d = 1$ , implying that all  $w_i > 0$ , in other words all participating trust scores are taken into account.

### 2.3. Evaluation of trust metrics

The performance of a given  $(Prop, Sel, Agg)$  triple is typically tested using leave-one-out experiments: in each such experiment, a directly available trust score  $(t, d)$  between a pair of agents  $x$  and  $y$  is removed from the trust network, and the given trust estimation strategy is used on the remaining trust network to predict this trust score. The generated trust estimations can then be evaluated by means of their accuracy. To measure accuracy, we may use the following two variations [14] on mean absolute error (MAE) and root mean squared error (RMSE), with  $(t_1, d_1), \dots, (t_p, d_p)$  and  $(t'_1, d'_1), \dots, (t'_p, d'_p)$  the original and estimated trust scores, respectively ( $T-MAE \in [0, 2]$  and  $T-RMSE \in [0, \sqrt{2}]$ ):

$$T-MAE = \frac{\sum_{i=1}^p |t_i - t'_i| + |d_i - d'_i|}{p} \quad T-RMSE = \sqrt{\frac{\sum_{i=1}^p (t_i - t'_i)^2 + (d_i - d'_i)^2}{p}}$$

<sup>4</sup> As discussed in [14], this goes at the expense of some of the boundary conditions.

<sup>5</sup> The Ordered Weighted Average (OWA, [21]) operator associates weights to the ordered positions of the scalar values  $\langle a_1, \dots, a_n \rangle$  that have to be aggregated. Let  $W = \langle w_1, \dots, w_n \rangle$  a weight vector such that  $\forall i \in \{1, \dots, n\}, w_i \in [0, 1]$  and  $\sum_{i=1}^n w_i = 1$ . Then the OWA-operator associated with  $W$  is defined as  $OWA_W(a_1, \dots, a_n) = \sum_{i=1}^n w_i b_i$ , where  $b_i$  is the  $i$ -th largest element in  $\langle a_1, \dots, a_n \rangle$ .

Another commonly used evaluation criterion for trust metrics is coverage, which refers to the fraction of trust scores for which a prediction can be generated. In our setting, coverage is directly influenced by the selection operator  $Sel$ ; indeed, some agents may be linked through the network, such that in principle it is possible to generate a trust estimation for them, but if the selection step eliminates all paths between them, the coverage of the corresponding trust estimation strategy will be reduced.

Note that coverage and accuracy should always be considered together: it makes no sense to have an algorithm that can perfectly predict trust scores if its corresponding coverage is unacceptably low, or vice versa. Ideally, a trust score estimation strategy should have maximal coverage and minimal T-MAE and T-RMSE.

### 3. Path length incorporation strategies

A problem that arises when aggregating trust scores is that some trust scores might be more reliable than others because they were aggregated along shorter paths. This was also argued by Golbeck [2] in the case of trust-enhanced recommender systems; she therefore chooses to consider only shortest paths linking two agents; in a similar vein, Massa and Avesani [9] only take into account paths shorter than a predefined threshold length. In this section, we propose a number of refinements to these strategies in order to make better use of the information contained in longer paths.

#### 3.1. Aggregation operators dependent on path length

Consider  $n$  trust scores  $(t_1, d_1), \dots, (t_n, d_n)$ , all different from  $(0, 0)$ , obtained by propagating  $p_1, \dots, p_n$  trust scores, respectively. We call  $p_i$  the *path length* of the trust score.

##### 3.1.1. Path length dependent weighted average

A first method consists of defining weights that depend on the path length of the propagated trust scores: trust scores propagated along longer paths get lower weights than trust scores propagated along shorter paths. Suppose we have such a path length weight vector  $W_p$ , then we can define a weighted average<sup>6</sup> trust aggregator  $P-WA$ :

$$P-WA((t_1, d_1), \dots, (t_n, d_n)) = (WA_{W_p}(t_1, \dots, t_n), WA_{W_p}(d_1, \dots, d_n)).$$

An easy way to obtain  $W_p$  is to define it as  $W_p = \langle w_1, \dots, w_n \rangle$ , such that  $\forall i, j = 1, \dots, n$ ,  $w_i = \frac{1}{\sum_{j=1}^n \frac{1}{p_j}}$ . However, this approach has the disadvantage of being very sensitive to the number of paths connecting two agents, as the following example illustrates.

**Example 1.** *Suppose there are 2 paths of length 2 and 9 paths of length 3. Then both paths of length 2 will get a weight 1/8, and the 9 paths of length 3 each get weight 1/12. The difference between the weights is small, so shorter paths receive little benefit.*

---

<sup>6</sup> Let  $W = \langle w_1, \dots, w_n \rangle$  a weight vector such that  $\forall i \in \{1, \dots, n\}$ ,  $w_i \in [0, 1]$  and  $\sum_{i=1}^n w_i = 1$ . Then the WA-operator associated with  $W$  is defined by  $WA_W(a_1, \dots, a_n) = \sum_{i=1}^n w_i a_i$ .

The advantage of being on a shorter path diminishes as the number of trust scores on longer paths increases. Since it can usually be expected that there will be much more longer than shorter paths, this is a serious drawback. To alleviate this problem we introduce a parameter  $\alpha > 1$  in order to penalize higher path lengths, defining  $w_i = \frac{\frac{1}{p_i^\alpha}}{\sum_{j=1}^n \frac{1}{p_j^\alpha}}$ . The corresponding path length dependent aggregation operator is called  $P-WA_\alpha$ .

**Example 2.** Suppose  $\alpha = 2$ . In the case of Example 1, the paths of length 2 now get a weight of  $1/6$  while those of length 3 get a weight of  $2/27$ . This is clearly an improvement, as a path of length 2 now counts for more than the double of a path of length 3.

Another solution is to take into account  $N(p)$ , the number of paths of length  $p$ . Specifically, we first compute a global weight  $T(p)$  attributed to paths of length  $p$  by the recursive formula ( $p = 2, 3, \dots$ )<sup>7</sup>

$$T(p) = \left( \frac{N(p)}{N(p) + 1} \right)^\alpha * \left( 1 - \sum_{q=2}^{p-1} T(q) \right)$$

This formula expresses that the larger  $N(p)$ , the less weight remains for trust scores on longer paths. The parameter  $\alpha \geq 0$  can be tuned to either strengthen ( $\alpha < 1$ ) or weaken ( $\alpha > 1$ ) this effect. The weight vector  $W_p = \langle w_1, \dots, w_n \rangle$  is then finally computed by

$$\forall i \in \{1, \dots, n\} : w_i = \frac{\frac{T(p_i)}{N(p_i)}}{\sum_{j=1}^n \frac{T(p_j)}{N(p_j)}}$$

We call the corresponding path length dependent aggregation operator  $P-WA_\alpha^*$ .

**Example 3.** Suppose  $\alpha = 1$ . For the data of Example 1,  $T(2) = 2/3$  and  $T(3) = 3/10$ . As a result, the paths of length 2 will get weight  $20/121$  and the nine paths of length 3 each get weight  $9/121$ ; these weights are similar to the ones obtained in Example 2.

### 3.1.2. Path length dependent ordered weighted average

The path length weight vector  $W_p$  can also be combined with the OWA-based aggregator K-OWA from the previous section. Note that path length weights  $W_p$  are associated with fixed trust score arguments, whereas the OWA weights  $W_{m_t}$  and  $W_{m_a}$  correspond to ordered positions. To deal with this, we introduce the combined OWA operator  $OWA_{U,V}^c$ , with OWA weight vectors  $U = \langle u_1, \dots, u_n \rangle$  and  $V = \langle v_1, \dots, v_n \rangle$ . Given a vector  $\langle a_1, \dots, a_n \rangle$  of scalar values to aggregate, we can always find a permutation  $\pi$  of  $\{1, \dots, n\}$  such that  $a_{\pi(1)} \geq \dots \geq a_{\pi(n)}$ . We then define the combined weight vector  $W = \langle w_1, \dots, w_n \rangle$  by  $w_i = \frac{u_i v_{\pi(i)}}{\sum_{i=1}^n u_i v_{\pi(i)}}$ , and put  $OWA_{U,V}^c = OWA_W$ . Applying

<sup>7</sup> Note that this formula can also be expressed using primitive recursion: for  $p = 2, 3, \dots$ ,

$$T(p+1) = \kappa_{p+1} * T(p) * \frac{1 - \kappa_p}{\kappa_p}$$

where  $T(2) = \kappa_2$  and  $\kappa_p = \left( \frac{N(p)}{N(p)+1} \right)^\alpha$ .

this operator to combine the path length weight vector  $W_p$  and the OWA weight vectors  $W_{m_t}$  and  $W_{m_d}$ , we obtain *KP-OWA*:

$$KP\text{-}OWA_{m_t, m_d}((t_1, d_1), \dots, (t_n, d_n)) = (OWA_{W_t, W_p}^c(t_1, \dots, t_n), OWA_{W_d, W_p}^c(d_1, \dots, d_n))$$

Different parameters  $m_t$  and  $m_d$  for trust and distrust can be chosen, but the path length weights are equal for the trust and distrust component.

### 3.1.3. Induced OWA path length incorporation

Another strategy to incorporate path length is to use *IOWA* operators<sup>8</sup>. Since the order inducing variables appearing in the *IOWA* approach do not need to be scalar (as long as a suitable linear order can be imposed upon them), further flexibility can be introduced in the trust aggregation process. In particular, we consider the combination of trust/distrust degrees and path length as inducing variables. Let  $V_t = \langle v_{t1}, \dots, v_{tn} \rangle$  and  $V_d = \langle v_{d1}, \dots, v_{dn} \rangle$ , such that  $v_{ti} = (p_i, t_i)$ ,  $v_{di} = (p_i, d_i)$ , and define the linear order  $\leq_p$  by  $(p_1, r_1) \leq_p (p_2, r_2) \Leftrightarrow (p_1 > p_2) \vee (p_1 = p_2 \wedge r_1 \leq r_2)$ . Using this definition we can define the path length incorporating aggregator *P-IOWA* as follows:

$$P\text{-}IOWA_{m_t, m_d}((t_1, d_1), \dots, (t_n, d_n)) = \begin{aligned} & (IOWA_{W_{m_t}}(t_1, v_{t1}), \dots, (t_n, v_{tn}), \\ & IOWA_{W_{m_d}}(d_1, v_{d1}), \dots, (d_n, v_{dn})) \end{aligned}$$

The idea of *P-IOWA* is that trust scores are ordered according to their path length, and then fixed decreasing weights are associated to these ordered positions such that trust scores propagated along shorter paths get higher weights.

### 3.2. Dynamic horizon search strategy

As mentioned before, when paths from  $x$  to  $y$  are searched, an upper limit called horizon will be imposed on the length of these paths, both for reasons of computational complexity and in order to avoid excessively long paths. The choice of the horizon is crucial for the effectiveness of aggregation. If  $h$  is chosen too small, the number of agent pairs for which a trust estimation can be made may be reduced drastically (low coverage); on the other hand, if  $h$  is too high, the presence of a large number of long paths may muddle the predicted trust score (low accuracy).

To tackle this problem, instead of using a fixed horizon, we propose a path search strategy involving a dynamically changing horizon. This is set, for a pair of agents, to the length  $s$  of the shortest path connecting them, as long as  $s$  does not surpass a global horizon  $h$ . In other words, only minimal length paths between  $x$  and  $y$  are considered, provided  $s \leq h$ . Note that the dynamic horizon search strategy coincides with the path length dependent weighted average aggregator *P-WA*<sub>0</sub><sup>\*</sup> from Section 3.1.1: indeed, in this case  $T(p) = 1$  if  $p$  is equal to the length of the shortest path, and  $T(p) = 0$  otherwise.

<sup>8</sup> The Induced Ordered Weighted Average (*IOWA*, [22]) operator associates weights to the ordered positions of the values of an order inducing variable. Let  $W = \langle w_1, \dots, w_n \rangle$  be an OWA weight vector, and  $\langle v_1, \dots, v_n \rangle$  a vector of values drawn from a linearly ordered space  $(V, \leq_V)$ . Then the *IOWA*-operator associated with  $W$  is defined by  $IOWA_W(\langle a_1, v_1 \rangle, \dots, \langle a_n, v_n \rangle) = \sum_{i=1}^n w_i b_i$  where  $b_i = a_j$  iff  $v_j$  is the  $i$ -th largest element in  $\{v_1, \dots, v_n\}$ .



The fact that all considered paths have the same length also implies that the path length dependent trust aggregators from Section 3.1 normally cannot be used in conjunction with this search strategy. It is however possible to combine the ideas behind both path length incorporating strategies in a meaningful way: if the number of trust scores  $(t_i, d_i)$  for which the path length  $p_i$  is larger than a given threshold  $t$ , then only these paths are considered. In the opposite case, all paths for which the length does not exceed the fixed horizon  $h$  are considered. In other words, in this semi-dynamical strategy we work only with the shortest paths, provided there are enough of them.

#### 4. Experimental Evaluation

In this section, we evaluate the path length incorporating aggregation strategies from the previous section, and compare them to their classical counterparts from [14]. Table 1 lists all aggregators used in the experiments. Apart from the 4 basic operators *T-MAX*, *D-MAX*, *K-MAX* and *K-OWA*, we also used the classical average *AV*, which serves as a baseline for the path length dependent weighted average *P-WA*.

<i>T-MAX</i>	Maximizes trust degree for given knowledge degree
<i>D-MAX</i>	Maximizes distrust degree for given knowledge degree
<i>K-MAX</i>	Maximizes knowledge degree
<i>K-OWA</i>	Extends <i>K-MAX</i> using OWA weight vector
<i>AV</i>	Computes classical average of trust and distrust degrees
<i>P-WA</i>	Computes weighted average of trust and distrust degrees based on path length dependent weight vector
<i>KP-OWA</i>	Extends <i>K-OWA</i> using path length dependent weight vectors
<i>P-IOWA</i>	Uses path length dependent order to rank trust scores before applying OWA weight vector to them

Table 1: Overview of trust aggregators used in experiments

##### 4.1. Data Set

The data set we used in our experiments was obtained from CouchSurfing.org, an organisation that offers travelers a platform to offer and find an overnight place (couch) to stay.<sup>9</sup> Trust is an important factor for this organisation, as people only want to offer or accept a couch from people they can trust. Registered users of CouchSurfing can fill out an online form where they can express, using two linguistic scales, how much they trust each other and how well they know each other. In [14], a heuristic method to map this information into trust scores was proposed: first, the trust and knowledge statements are translated into  $[0, 1]$  as in Table 2. The available trust and knowledge information are then mapped to trust scores according to the following formula:  $(t, d) = (k \cdot t', k \cdot (1 - t'))$ , with  $t'$  ( $k$ ) the translation of the trust (knowledge) statement<sup>10</sup>. The resulting trust network consists of 397 471 users and 2 697 705 trust statements.

<sup>9</sup> The data set can be obtained from the authors on request, pending consent of CouchSurfing.org.

<sup>10</sup> Records that contain a ‘not at all’ knowledge statement or a ‘don’t know well enough to decide’ trust statement are translated to  $(0, 0)$ .

Table 2: Translation of trust and knowledge statements in the CouchSurfing data set.

<i>trust statements</i>		<i>knowledge statements</i>	
<i>label</i>	<i>t'</i>	<i>label</i>	<i>k</i>
don't know	0	not at all	0
don't trust	0	a little bit	0.25
somewhat trust	0.25	somewhat	0.5
generally trust	0.5	fairly well	0.75
highly trust	0.75	very well	1
trust with my life	1	extremely well	1
		couldn't know any better	1

#### 4.2. Setup

To evaluate the different strategies, we performed a series of 10 000 leave-one-out experiments, in which each time we remove a trust score  $(t, d)$  from the trust network and try to predict it using a given aggregator and search strategy (fixed, or dynamic horizon). The propagation strategy is fixed, in all cases, to the *Prop* operator defined in Section 2. We used only horizons  $h = 2$  and 3, since larger horizons incur computational problems for this dataset. For  $h = 3$ , we also tested dynamic horizon (only shortest paths are considered) and semi-dynamic horizon with  $t = 1$  (only shortest paths are considered, provided there is more than one such path). Note that if  $h = 2$ , the dynamic horizon strategy coincides with the fixed horizon strategy, since all considered paths have the same length.

As mentioned in Section 2, trust estimations are evaluated by means of coverage and accuracy. Whereas in our setup coverage depends only on the used horizon (regardless of whether it is fixed or dynamic), different strategies normally lead to different T-MAE and T-RMSE values; apart from comparing their absolute values, we also use Wilcoxon's signed-rank statistical test [18], with a significance level  $\beta = 0.01$ , to be able to verify whether the observed differences are significant.

#### 4.3. Results

First, we compare the behaviour of the basic trust aggregators introduced in [14] in the situation when paths of maximal length  $h = 2$ , resp.  $h = 3$  are considered.<sup>11</sup> In the former case, the coverage is 57.46%: this means that for 5746 out of the 10 000 trust scores, at least one path of length 2 exists, which allows a prediction to be made. When  $h = 3$ , more paths become available and the coverage rises to 66.83%.

However, this increase in coverage also leads to a general decrease in accuracy, as can be seen in Fig. 2 which shows the corresponding T-RMSEs and T-MAEs for the different trust aggregators. Note that the difference is smallest for *K-OWA*, which also performs the best among the basic operators ( $T$ -RMSE = 0.2972 and  $T$ -MAE = 0.2893 for  $h = 2$ ; and  $T$ -RMSE = 0.2977 and  $T$ -MAE = 0.2943 for  $h = 3$ ).

<sup>11</sup> Note that [14] only studied the case  $h = 2$ .

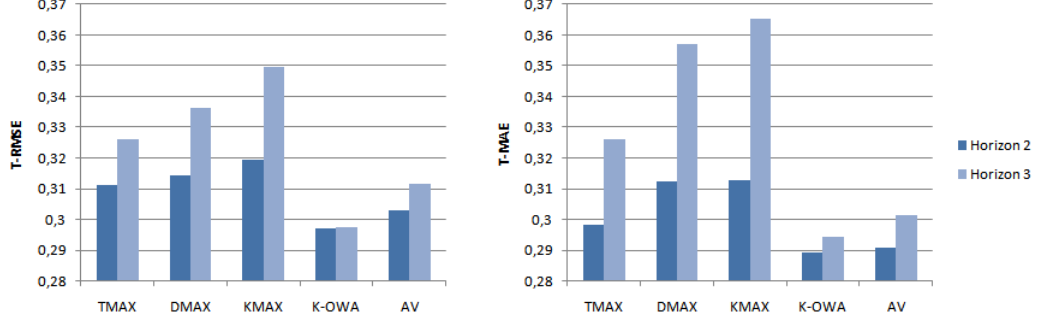


Figure 2:  $T$ -RMSE and  $T$ -MAE for basic, path length independent, trust aggregators for  $h = 2$  and  $3$ .

Next, we study the behaviour of the path length dependent trust aggregators. Note that when  $h = 2$ ,  $P-WA = AV$ , and  $KP-OWA = P-IOWA = K-OWA$ , i.e., path length incorporation has no effect. Hence, we focus on their effect when  $h = 3$ . In Fig. 3,  $T$ -RMSEs and  $T$ -MAEs are shown for the weighted average operators  $P-WA_\alpha$  and  $P-WA_\alpha^*$ , and compared to AV. From these graphs, it can be inferred that the simple AV can be improved by using the  $P-WA$  operators. For  $P-WA_\alpha$ , better results are obtained for higher  $\alpha$ , with  $\alpha = 10$  being close to the optimum ( $T$ -RMSE = 0.3012,  $T$ -MAE = 0.2910). On the other hand,  $P-WA_\alpha^*$  performs better for smaller  $\alpha$ , but slightly deteriorates when  $\alpha < 0.1$ . Overall, its results are slightly better than  $P-WA_\alpha$ , with the optimum for  $\alpha = 0.1$  ( $T$ -RMSE = 0.2992,  $T$ -MAE = 0.2902).

These observations confirm that it is a good idea to attribute a lot of weight to the shortest paths, but also to take into account the number of those shortest paths.

Similar observations can be made for the results of  $KP-OWA$  vs.  $K-OWA$  in Fig. 4. In this case the optimal value of  $\alpha$  is a bit smaller for the variant  $KP-OWA_\alpha$  ( $\alpha = 6$ ;  $T$ -RMSE = 0.2942 and  $T$ -MAE = 0.2891) and a bit higher for  $KP-OWA_\alpha^*$  ( $\alpha = 0.3$ ;  $T$ -RMSE = 0.2925 and  $T$ -MAE = 0.2886). The results are slightly better than those obtained with their  $P-WA$  counterparts; this demonstrates that the benefits of  $K-OWA$ 's fixed weight vector are further enhanced by introducing path length dependent weights.

The figure also shows the results for  $P-IOWA$ . Unfortunately, neither its  $T$ -RMSE (0.3061) nor  $T$ -MAE (0.2999) improves that of  $K-OWA$ . So, the path length dependent order and the path length independent weight vector do not reinforce each other.

Globally, the  $T$ -MAE and  $T$ -RMSE differences between the classical strategies AV and  $K-OWA$  and their path length incorporating extensions are small. However, the Wilcoxon test confirms that the differences AV vs.  $P-WA_{10}$ , AV vs.  $P-WA_{0.1}^*$ ,  $K-OWA$  vs.  $KP-OWA_6$ , and  $K-OWA$  vs.  $KP-OWA_{0.3}^*$  are all significant at the 0.01 level; in other words, our proposed strategies consistently outperform the classical ones.

In order to get a better understanding of the types of trust score estimations that benefit from path length incorporation, in the left hand graph in Fig. 5, we show a split-up of the  $T$ -RMSE results for  $K-OWA$  and  $KP-OWA_\alpha^*$  ( $\alpha = 0.3$ ) according to the number  $n$  of paths leading from the source to the target agent (i.e., the number of trust scores

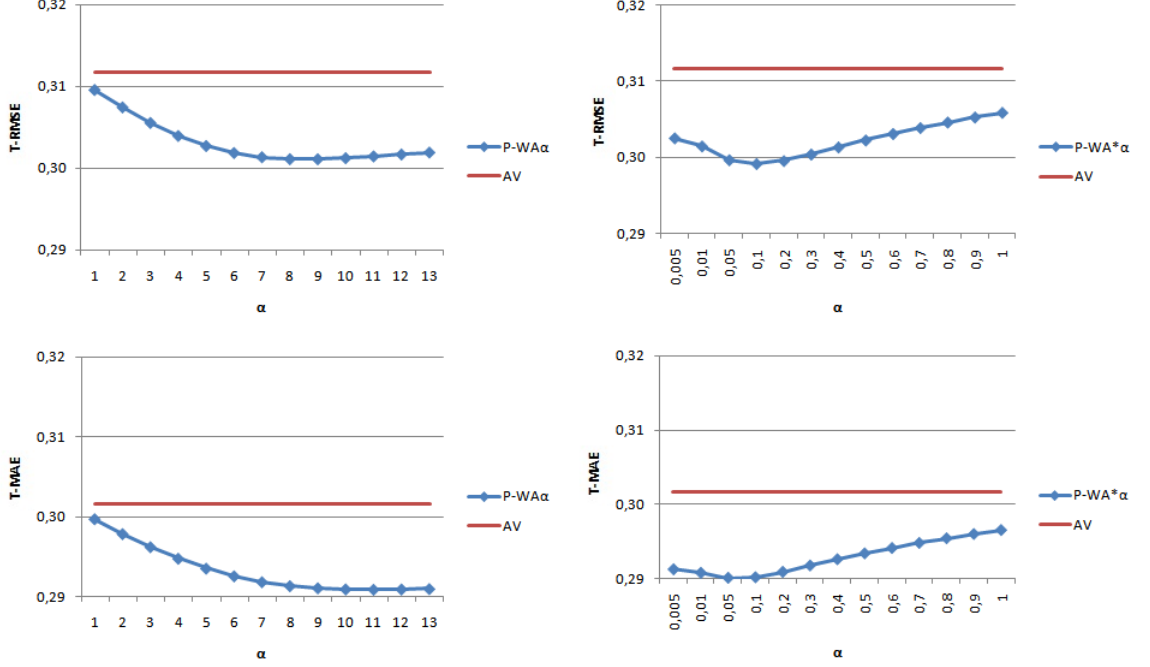


Figure 3:  $T$ -RMSE and  $T$ -MAE for path length dependent weighted average operators  $P-WA_\alpha$  and  $P-WA_\alpha^*$ , compared to the classical average  $AV$ , for  $h = 3$ .

to be aggregated)<sup>12</sup>. From this, we learn that for small values of  $n$ , the two approaches differ very little, but as  $n > 10$ , a more or less constant difference can be observed. For the latter group of experiments, we made a further distinction based on the ratio  $r = \frac{N(3)}{N(2)+N(3)}$ , i.e., the fraction of paths that have length 3. In the right hand graph of Fig. 5, we show the difference between  $K-OWA$  and  $KP-OWA_\alpha^*$ 's  $T$ -RMSEs according to this split-up. We can infer that the lower  $r$ , the higher the observed difference; this means that  $KP-OWA_\alpha^*$  performs better when there are more paths with different length to aggregate. It can also be seen that only a very small portion of the data have less than 4 times as many paths of length 3 than paths of length 2.

Next, we study the effect of using a dynamic horizon search strategy in Fig. 6. It can be seen that this path length incorporating strategy is beneficial for all basic trust aggregators. This confirms the importance of using shorter paths when they are available.

Comparing with Fig. 2, the  $T$ -RMSE results of all operators for dynamic horizon 3 are slightly better even than those for fixed horizon 2; this is a valuable result, which shows that coverage and accuracy can be increased simultaneously. The overall best result with dynamic search is obtained for  $K-MAX$ :  $T$ -RMSE = 0.2970 and  $T$ -MAE = 0.2896.

<sup>12</sup> Similar observations can be made for the  $T$ -MAEs, as well as for the other operators we considered.

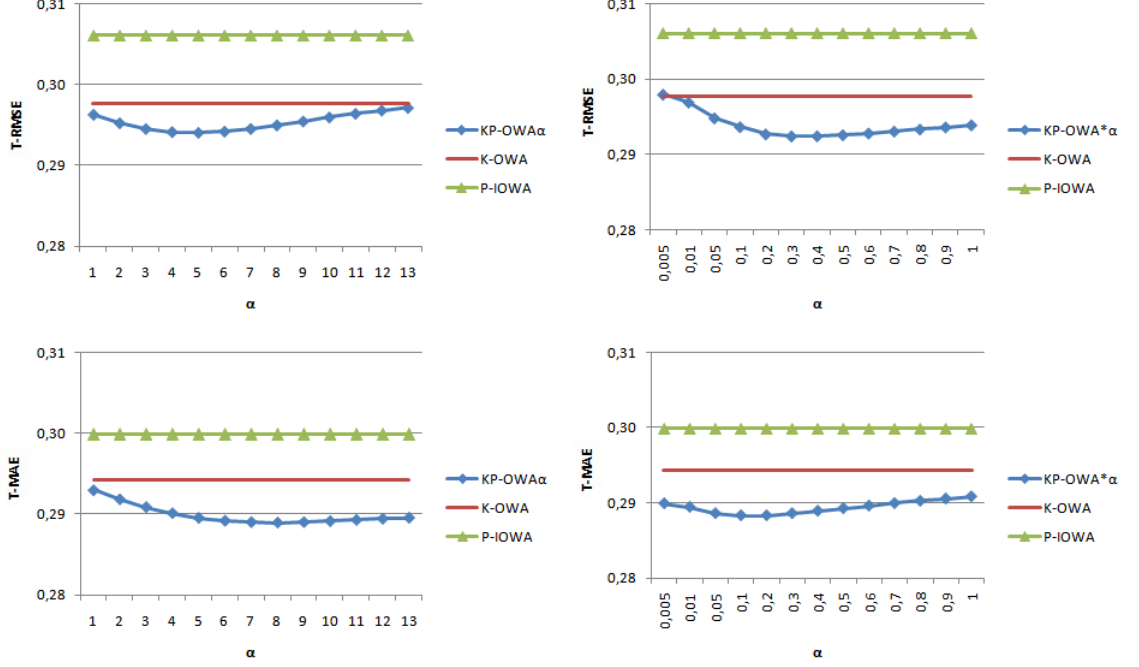


Figure 4:  $T$ -RMSE and  $T$ -MAE for path length dependent (induced) ordered weighted average operators  $KP\text{-OWA}_\alpha$  and  $KP\text{-OWA}_\alpha^*$  and  $P\text{-IOWA}$ , compared to  $K\text{-OWA}$ , for  $h = 3$ .

Finally, we investigate whether the advantages of both path length incorporating strategies can be combined by using the semi-dynamic search strategy. To this aim, in Fig. 7, we compare  $K\text{-OWA}$  to  $KP\text{-OWA}_\alpha^*$ , the best of the path length dependent trust aggregators.  $K\text{-OWA}$  is evaluated using dynamic horizon  $h = 3$ , while for  $KP\text{-OWA}_\alpha^*$  we use both a fixed horizon  $h = 3$ , as well as a semi-dynamical horizon  $h = 3$  with threshold  $t = 1$ . In practice the latter strategy implies the following:

- If there is exactly one path of length 2 connecting two agents, then  $KP\text{-OWA}_\alpha^*$  is executed in the same way as with a fixed horizon. In particular, all paths of length 3 are used in the calculation and get a lower weight than the single length 2 path.
- In the other cases, only shortest paths are considered. Since all paths used in the calculation have equal length,  $KP\text{-OWA}_\alpha^*$  reduces to  $K\text{-OWA}$  in this case.

The graphs show that a fixed horizon  $KP\text{-OWA}_\alpha^*$  already obtains lower  $T$ -RMSE and  $T$ -MAE than dynamic  $K\text{-OWA}$  over part of  $\alpha$ 's range. However, the Wilcoxon test reveals that even for the optimal value  $\alpha = 0.3$  these differences are not statistically significant. On the other hand, when a semi-dynamic horizon is used, the results can be further improved, and in this case the  $T$ -RMSE and  $T$ -MAE of  $KP\text{-OWA}_\alpha^*$  are significantly better than those of dynamic  $K\text{-OWA}$  according to the Wilcoxon test.

The benefit of using the semi-dynamic horizon strategy can also be seen in Fig.8, which shows the split-up of  $KP\text{-OWA}_{0.3}^*$ 's results according to the number of paths

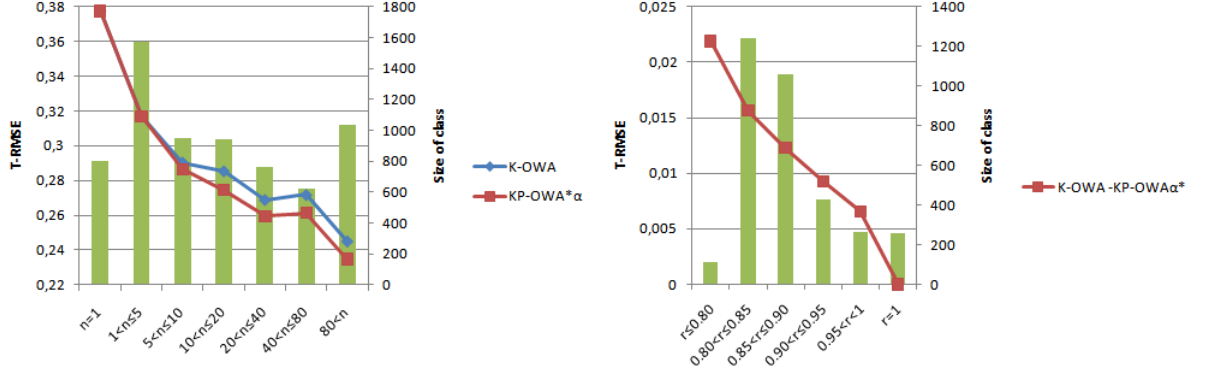


Figure 5: a)  $T$ -RMSE for  $K$ -OWA and  $KP$ -OWA\* $\alpha$  ( $\alpha = 0.3$ ) for  $h = 3$ , according to the number  $n$  of aggregated trust scores; b) difference between  $T$ -RMSE for  $K$ -OWA and  $KP$ -OWA\* $\alpha$  ( $\alpha = 0.3$ ) for  $h = 3$ , according to the ratio of paths of length 3 to the total number of paths. The bars denote the number of elements in each class.

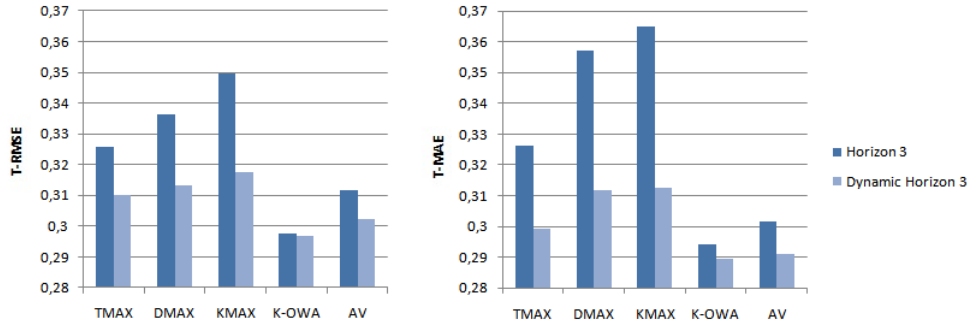


Figure 6:  $T$ -RMSE and  $T$ -MAE for basic, path length independent, trust aggregators for fixed and dynamic horizon  $h = 3$ .

of length 2 used in the aggregation. It can be seen that if there is exactly one path of length 2, using the dynamic horizon more often results in large errors (larger  $T$ -RMSE) than using the fixed or semi-dynamic strategies, which are equal in this case. On the other hand, if there are several 2-length paths, it is better to use dynamic horizon (equal to the semi-dynamic strategy in this case) than fixed horizon. This gives us a very nice proof that a thoughtful use of path length incorporating strategies can yield more, and more accurate trust score predictions.

## 5. Conclusion

In this paper, we have introduced several aggregation strategies for trust scores with variable path lengths. Since less propagation errors are involved in trust scores propagated along shorter paths, they should be considered more important than other trust scores during the trust aggregation process.

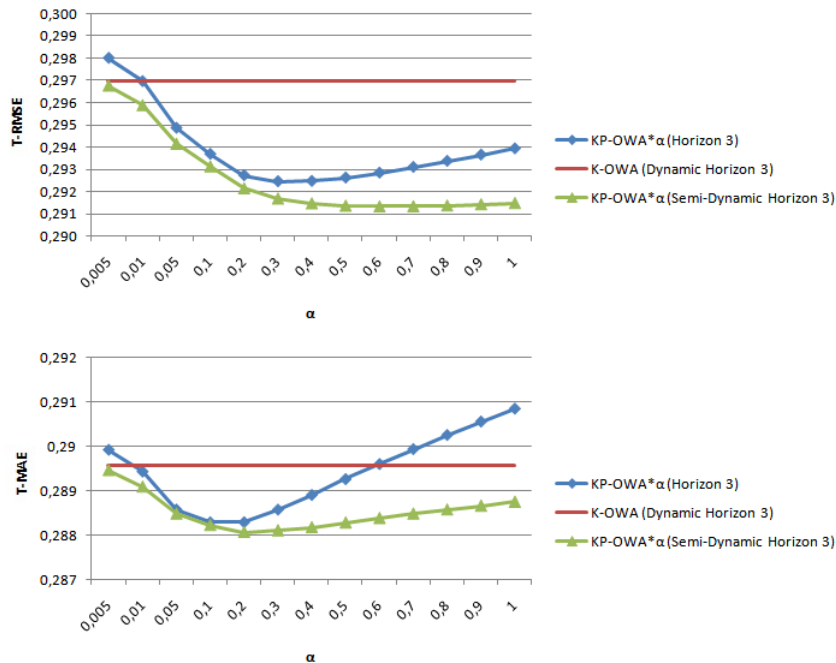


Figure 7:  $T$ -RMSE and  $T$ -MAE for path length independent  $K$ -OWA with dynamic horizon  $h = 3$ , and path length dependent  $KP$ -OWA $_{\alpha}^*$  with fixed and semi-dynamic horizon  $h = 3$ .

In order to make this distinction, we have introduced two general types of methods. First, we attach weights to trust scores that depend on the number of propagations needed to obtain them. We have defined such weights on the basis of the inverses of the path lengths of the trust scores, such that trust scores propagated along longer paths get lower weights. As this approach is sensitive to the number of trust scores that have a given path length, we have also introduced a method that reserves a total weight for all trust scores with a certain path length, dependent on the number of trust scores of this path length. Both approaches can be enhanced with a factor that determines to which extent path length is taken into account.

Secondly, we have introduced a dynamic horizon search strategy: trust scores with a certain path length are dismissed whenever there are others available that were propagated along shorter paths. We also introduced a modification of this method, the semi-dynamic horizon search strategy, that only considers trust scores of higher path lengths if the number of trust scores with shortest path length does not exceed a threshold.

We have performed an experimental study on the CouchSurfing trust network, which confirmed that trust scores propagated along longer paths indeed contain more errors than others. Furthermore, we learned that both using path length weights and using a dynamical horizon search strategy improve the aggregation result. Path length weights that depend on the number of trust scores of each path length lead to the best results. On the other hand, the semi-dynamic horizon strategy can be combined with trust aggregators that use path length depending weights. The results for this combination are

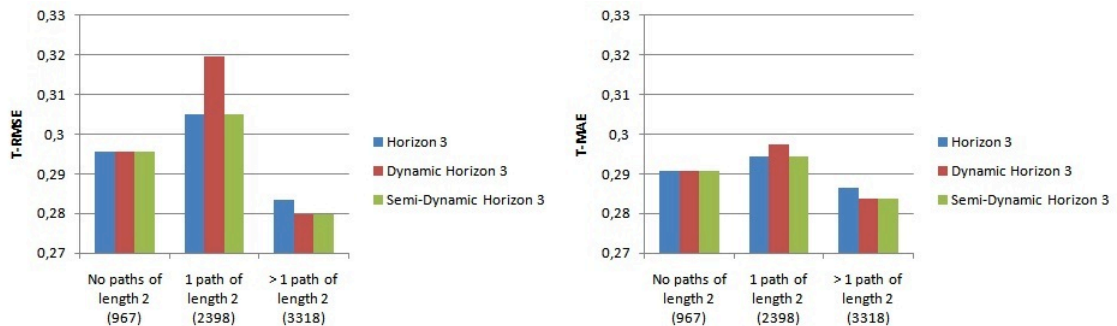


Figure 8:  $T$ -RMSE and  $T$ -MAE for path length independent  $K$ -OWA with dynamic horizon  $h = 3$ , and path length dependent  $KP$ -OWA $_{\alpha}^*$  with fixed and semi-dynamic horizon  $h = 3$ . The numbers in brackets on the  $X$  axis denote the number of elements in each class.

the best ones overall.

As part of our future work, we will consider the refinement of trust aggregators using other information about the paths that connect agents besides their length. For example, the quality of a path is also determined by the authority each node on the path enjoys.

## Acknowledgements

Chris Cornelis would like to thank the Research Foundation–Flanders for funding his research. This paper has also been developed with the financing of FEDER funds in TIN2010-17876, TIC-5299 and TIC-05991.

## 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] J. Golbeck and J. Hendler. Filmtrust: Movie recommendations using trust in web-based social networks. In *Proceedings of the IEEE Consumer Communications and Networking Conference*, pages 282–286, 2006.
- [4] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proc. of WWW*, pages 403–412, 2004.
- [5] A. Jøsang. Trust-based decision making for electronic transactions. In *Proceedings of the Fourth Nordic Workshop on Secure IT Systems*, 1999.
- [6] A. Jøsang and S. J. Knapskog. A metric for trusted systems. In *In Proceedings of the 21st National Security Conference. NSA*, pages 16–29, 1998.
- [7] A. Jøsang, S. Marsh, and S. Pope. Exploring different types of trust propagation. In *Lecture Notes in Computer Science 3986*, pages 179–192, 2006.
- [8] U. Kuter and J. Golbeck. Sunny: a new algorithm for trust inference in social networks using probabilistic confidence models. In *Proc. of AAAI*, pages 1377–1382, 2007.
- [9] P. Massa and A. Avesani. Trust-aware recommender systems. In *Proc. of the ACM Recommender Systems Conference*, pages 17–24, 2007.
- [10] P. Massa and P. Avesani. Trust metrics in recommender systems. In *Computing with Social Trust*, pages 259–285. 2009.
- [11] M. McPherson, L. Smith-Lovin, and J. M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.



- [12] J. Sabater-Mir and M. Paolucci. On representation and aggregation of social evolutions in computational trust and reputation models. *International Journal of Approximate Reasoning*, 46(3):458–483, 2007.
- [13] P. Victor. *Trust networks for recommender systems*. PhD thesis, 2010.
- [14] P. Victor, C. Cornelis, M. De Cock, and E. Herrera-Viedma. Practical aggregation operators for gradual trust and distrust. *Fuzzy Sets and Systems*, 184(1):126–147.
- [15] 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.
- [16] A. Wierzbicki. *Trust and fairness in open, distributed systems, Studies in Computational Intelligence, vol. 27*. Springer-Verlag, 2010.
- [17] G. Wierzowiecki and A. Wierzbicki. Efficient and correct trust propagation using closelook. In *Proceedings of the IEEE/ACM Conference on Web Intelligence*, pages 676–681, 2010.
- [18] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1:80–83, 1945.
- [19] R. Yager. On the determination of strength of belief for decision support under uncertainty - part i: generating strengths of belief. *Fuzzy Sets and Systems*, 142(1):117–128, 2004.
- [20] R. Yager. On the determination of strength of belief for decision support under uncertainty - part ii: fusing strengths of belief. *Fuzzy Sets and Systems*, 142(1):129–142, 2004.
- [21] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man, and Cybernetics*, 18:183–190, 1988.
- [22] R.R. Yager and D. Filev. Induced ordered weighted averaging operators. *IEEE Transactions on Systems, Man, and Cybernetics*, 29(2):141–150, 1999.
- [23] C.-N. Ziegler and G. Lausen. Propagation models for trust and distrust in social networks. *Information Systems Frontiers*, 7:337–358, 2005.