Gradual Trust and Distrust in Recommender Systems

Patricia Victor[a] Chris Cornelis[a] Martine De Cock[a]

[a]Computational Web Intelligence,
Dept. of Applied Mathematics and Computer Science, Ghent University
Krijgslaan 281 (S9), 9000 Gent, Belgium

Paulo Pinheiro da Silva[b]

[b]Dept. of Computer Science, The University of Texas at El Paso
Computer Science Building 222B, 500 W University Ave, El Paso, USA

Abstract

Trust networks among users of a recommender system (RS) prove beneficial to the quality and amount of the recommendations. Since trust is often a gradual phenomenon, fuzzy relations are the pre-eminent tools for modeling such networks. However, as current trust-enhanced RSs do not work with the notion of distrust, they cannot differentiate unknown users from malicious users, nor represent inconsistency. These are serious drawbacks in large networks where many users are unknown to each other and might provide contradictory information. In this paper, we advocate the use of a trust model in which trust scores are (trust,distrust)-couples, drawn from a bilattice that preserves valuable trust provenance information including gradual trust, distrust, ignorance, and inconsistency. We pay particular attention to deriving trust information through a trusted third party, which becomes especially challenging when also distrust is involved.

Key words: web of trust, trust propagation, recommender system, bilattice theory

Email addresses: `Patricia.Victor@UGent.be` (Patricia Victor), `Chris.Cornelis@UGent.be` (Chris Cornelis), `Martine.DeCock@UGent.be` (Martine De Cock), `paulo@utep.edu` (Paulo Pinheiro da Silva).

1 Introduction

Collaboration, interaction and information sharing are the main driving forces of the next generation of web applications referred to as 'Web 2.0' [18]. Well-known examples of this emerging trend include weblogs (online diaries or journals for sharing ideas instantly), Friend-Of-A-Friend [1] (FOAF) files (machine-readable documents describing basic properties of a person, including links between the person and objects/people they interact with), wikis (web applications such as Wikipedia [2] that allow people to add and edit content collectively) and social networking sites (virtual communities where people with common interests can interact, such as dating sites or car addict forums). In this paper, we focus specifically on recommender systems (RSs) [20], i.e. applications that are designed to suggest items (books, movies, web pages, travel packages, etc.) to users who might be interested in them, given some information about users' profiles and relationships between users. A classical example is the system employed by Amazon [3], which offers recommendations like "Customers who bought this item also bought ..."

Apart from opportunities, collaboration also brings some reasons of concern. As users can freely contribute new information, they affect application results in an unpredicted, potentially abusive manner. As such, adequate controls are required to warrant information quality and trustworthiness. An attractive solution, nicely exploiting the social dynamics that drives this new wave of web applications, is the deployment of trust networks: collections of agents (humans or machines) connected by trust relations indicating whether agents in these networks trust, or distrust, each other. Many researchers have recently turned to this topic, as witnessed by an increasing amount of publications in the area (see e.g. [13,21,24,27]); a particular focus of interest is on the development of gradual models that quantify the degree to which agents may trust each other [1,2,11,12,15,16]. These models reflect the fact that in real life, too, trusting someone is seldom a black-or-white phenomenon, and that people often trust each other "very much", "somewhat", ...

Trust networks can contribute to the success of RSs by allowing users to establish better-informed opinions about certain items through the judgment of trusted sources/agents that have evaluated or experienced those items. Trusted agents can make additional recommendations over the ones generated by other RS techniques, which especially benefit users who lack a properly detailed user profile. These recommendations may also achieve higher quality, as research [23] has pointed out that people tend to rely more on

[1] www.foaf-project.org
[2] www.wikipedia.org
[3] www.amazon.com

recommendations from people they trust, than on online RSs which generate recommendations based on anonymous people similar to them. Finally, trust networks can also be used to prevent malicious insiders from abusing the system to unnaturally boost some items' recommendability.

Although the incorporation of a trust model can alleviate some major RS issues, the existing trust-enhanced RSs still lack the ability to preserve important provenance information indicating how a suggested trust value has been derived. As such, users cannot really exercise their right to interpret how trust is computed, and moreover the quality of recommendations may be affected negatively. For example, in a system that cannot differentiate between absence of trust caused by presence of distrust (e.g., as towards a malicious agent) versus by lack of knowledge (e.g., as towards an unknown agent), it is much more difficult to detect malicious insiders, than in a system in which an explicit distinction is made between active distrust and ignorance. Moreover, in a system that cannot differentiate between arguments to half trust or half distrust a person, and (conflicting) arguments to simultaneously trust and distrust a person completely, users can draw the wrong conclusions.

This paper is organized as follows. In Section 2, we survey existing work on trust models and their usage in RSs. Section 3 introduces the bilattice-based trust model $\mathcal{BL}^{\square}$, in which the traditional trust degree is sided by a degree of distrust; we show that the model is able to represent gradual trust, distrust, ignorance and inconsistency simultaneously, as different but related concepts, and as such avoids some common pitfalls that single-valued models face. Besides advantages, our approach brings along some new challenges as well; in Section 4, we discuss possible strategies to support propagation of trust and distrust in RSs, i.e., the process of inquiring for a trust estimation with a trusted third party (TTP), who in turn might consult its own TTP, and so on. The paper is concluded by a discussion of subsequent problems that need to be addressed (Section 5).

2 Related work

Recommender system technologies and trust models constitute the two pillars of a trust-enhanced recommender system. We briefly discuss the former in the following subsection, and describe how trust can be used to improve them. We then proceed with classifying trust models to position the new approach that we introduce in Section 3, and explain why current models are not fully suitable for use in RSs.

2.1 Recommender systems

Content-based filtering (CB) [22] and collaborative filtering (CF) [19] are well-known examples of recommendation approaches. CB systems suggest items similar to the ones that the user previously liked. Such systems tend to have their recommendation scope limited to the immediate neighbourhood of the user's past purchase or rating record; for instance, if a customer of a video rental store has only ordered romantic movies, the system will continue to recommend just related items, and not explore other interests of the user. In this sense, RSs can be improved significantly by (additionally) using CF, which typically identifies users whose tastes are similar to those of the given user and recommends items that they have liked; ratings for unseen items are predicted based on a combination of the nearest neighbours' ratings. As an additional benefit over CB RSs, CF RSs do not require that the internal structure of the items be known, and consequently, can be applied to any domain.

Despite significant improvements on recommendation approaches, some important problems still remain. For instance, in the context of a CF RS, it is often difficult for the RS to find similar users, as users typically rate/experience only a small fraction of available items (the 'sparsity problem'). Moreover, it is challenging to generate good recommendations for users that are new to the system, as they have not rated a significant number of items and hence cannot properly be linked with similar users (the 'cold start' problem). Thirdly, because RSs are widely used in the realm of e-commerce, there is a natural motivation for producers of items (manufacturers, publishers, etc.) to abuse them so that their items are recommended to users more often [28]; a common 'copy-profile' attack consists in copying the ratings of the target user, which results in the system thinking that the adversary is most similar to the target.

In real life, a person who wants to avoid a bad deal may ask a friend, i.e., someone he trusts, what he thinks about a certain item i. If this friend does not have an opinion about i, he can ask a friend of his, and so on until someone with an opinion about i (i.e., a recommender) is found. Trust-enhanced RSs work in a similar way, as depicted in Fig. 1: once a path to a recommender is found, the RS can combine that recommender's judgment with available trust information to obtain a personalized recommendation. In this way, a trust network allows to reach more users and more items. In the CF RS in Fig. 2, agents a and b will be linked together because they have given similar ratings to certain items (among which i_1); analogously, b and c can be linked together. Consequently, a prediction of a's interest in i_2 can be made. But in this scenario there is no link between a (or b) and i_3 or, in other words, there is no way to find out whether i_3 would be a good recommendation for agent a.

This situation might change when a trust network has been established among

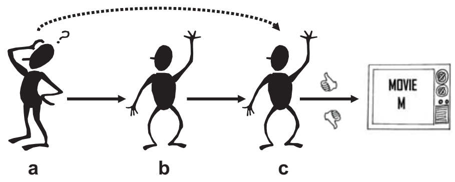

Fig. 1. Recommending an item

the users of the RS. Such trust networks can be generated automatically, or built by the explicit input of the users. Golbeck's FilmTrust [11] is an example of the latter type; it is an online social network combined with a movie rating and review system in which users are asked to evaluate their acquaintances on a scale from 1 to 10 according to their movie taste. FilmTrust is a non commercial venture, but trust-based systems are also being used in e-commerce applications like Epinions.com [4], a site that gives users the opportunity to include other users (based on their quality as reviewers) in their own 'web of trust'. A particular novelty of the Epinions system is that users can maintain a 'block list' of reviewers they explicitly distrust too. Including someone in his web of trust/block list corresponds (implicitly) with issuing a trust/distrust statement. These statements can be easily updated. An example of automatic generation of trust values can be found in [16], in which trust values are inferred from movie rating data, based on a user's history of making reliable recommendations.

Figures 2 and 3 illustrate the difference between a classical CF RS and a trust-enhanced RS. While in the former situation the RS is not able to generate a prediction about i_3 for user a, this could be solved in the latter: if a expresses a certain level of trust in b, and b in c, by propagation an indication of a's trust in c can be obtained. If it indicates that agent a may highly trust c, then i_3 might be a good recommendation for a, and will be highly ranked among the other recommended items. In this way, the sparsity problem is alleviated. In particular, as was found in [15], for new users, a few trust statements can already yield much higher recommendation coverage and reduced prediction error rates, hence alleviating the cold start problem. Moreover, a web of trust can be used to produce an indication about the trustworthiness of users and as such make the system less vulnerable to malicious insiders: a simple copy-profile attack will only be possible when the target user, or someone who is trusted by the target user, has indicated that he trusts the adversary to a certain degree.

It is clear that establishing a trust network among a RS's users can contribute to its success, and some attempts in this direction have already been made. Golbeck [11] investigates the role of trust ratings in FilmTrust. In [29], Ziegler describes the role of trusted neighbourhoods for RSs on the semantic web [5].

[4] `www.epinions.com`

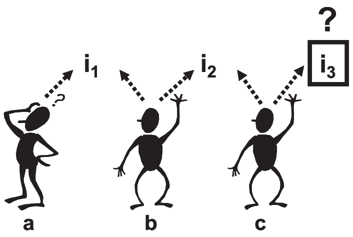

Fig. 2. Trust in RS (1)

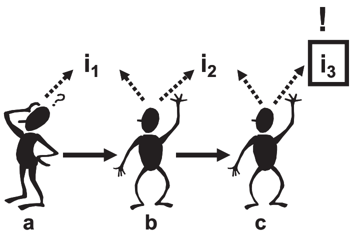

Fig. 3. Trust in RS (2)

O'Donovan and Smyth's work [16] focusses on automatically inferring trust based on a user's history, while Massa and Avesani [15] examine the effects of trust propagation through evaluation on a dataset obtained by crawling Epinions; however, no distrust information is taken into account in [15]. In fact, all mentioned approaches use a trust model which only considers trusted sources, and do not distinguish between distrusted and unknown sources.

The ability to propagate trust information is one of the main strengths of all these trust-enhanced RSs. The rationale behind it is based on the notion of trust transitivity: e.g., if a trusts b and b trusts c, then a trusts c (see e.g. [3,11]). Several types of trust transitivity can be distinguished, such as trust in an agent's competence to recommend a good item, or trust in an agent's competence to recommend/evaluate a good recommender agent (see e.g. [1]). Although these concepts are essentially different, most practical recommender applications assume a simplification of the 'real world': it is not always possible for a user a to know whether the scores/ratings he receives from a user b are made by b explicitely, or inferred via other TTPs, hence a cannot always distinguish between b as a recommender of good items and b as a recommender of good agents. In this respect, it might be more appropriate to speak of 'goodwill' or 'benevolence' instead of 'trust', but we chose to work with the latter term because it is the most accepted one in the trust/RS literature.

2.2 Trust models

Trust models come in many flavours and can be classified in several ways, among which probabilistic vs. gradual approaches as well as representations of trust vs. representations of both trust and distrust (see also [26]). This classification is shown in Table 1, along with some representative references for each class.

A *probabilistic* approach deals with a single trust value in a black or white fashion — an agent or source can either be trusted or not — and computes a probability that the agent can be trusted. Examples can, among others, be found in [13] in which Jøsang et al. present a model for quantifying and reasoning about trust in IT equipment, in [21] where a path algebra for computing

Table 1
Trust Models, State of the Art

	trust only	**trust and distrust**
probabilistic	Kamvar et al. [14] Richardson et al. [21] Zaihrayeu et al. [27]	Jøsang et al. [13]
gradual	Abdul-Rahman et al. [1] Falcone et al. [17] Almenárez et al. [2] Tang et al. [25] Golbeck [11] Massa et al. [15]	De Cock et al. [7] Guha et al. [12]

trust on the semantic web is proposed, or in contributions like Kamvar et al.'s Eigentrust algorithm [14] that focus on peer-to-peer (P2P) networks, or Zaihrayeu et al.'s question answering system IWTrust [27]. In such a setting, a higher suggested trust value corresponds to a higher probability that an agent can be trusted.

On the other hand, a *gradual* approach is concerned with the estimation of trust values when the outcome of an action can be positive to some extent, e.g. when provided information can be right or wrong to some degree, as opposed to being either right or wrong (e.g. [1,2,7,11,12,15,25]). Note that in real life, too, trust is often interpreted as a gradual phenomenon: humans do not merely reason in terms of 'trusting' and 'not trusting', but rather trusting someone 'very much' or 'more or less'. Fuzzy logic is very well-suited to represent such natural language labels which represent vague intervals rather than exact values. The last years witnessed a rapid increase of gradual trust approaches, ranging from socio-cognitive models (e.g. implemented by fuzzy cognitive maps in [17]), over management mechanisms for selecting good interaction partners on the web [25] or for pervasive computing environments (Almenárez et al.'s PTM [2]), to RS models [11,15] and general models for virtual communities [1,12].

Large agent networks without a central authority typically face ignorance and inconsistency problems. Indeed, it is unlikely that all agents know each other, and different agents might provide contradictory information. Both ignorance and inconsistency can have an important impact on the trust estimation. Models that only take into account trust (e.g. [1,2,14,15,27]), either with a probabilistic or a gradual interpretation, are not fully equipped to deal with trust issues in large networks where many agents do not know each other: in suggesting a trust value to an inquiring agent, valuable information on how this value has been obtained is lost. This is problematic, as user opinions may be affected by provenance information exposing how trust values have been computed. For example, a trust estimation of a source from a fully informed agent is quite different from a suggested trust value from an agent who does not know the source too well but has no evidence to distrust it.

A suitable solution to these problems are models that take into account both trust and distrust; [7,12,13]. To the best of our knowledge, there is only one probabilistic approach considering trust and distrust simultaneously: in Jøsang and Knapskog's subjective logic [13] (SL), an opinion includes a belief b that an agent is to be trusted, a disbelief d corresponding to a belief that an agent is not to be trusted, and an uncertainty u. The uncertainty factor leaves room for ignorance in this model. However, the requirement that the belief b, the disbelief d and the uncertainty u sum up to 1, rules out options for inconsistency even though this might arise quite naturally in large networks with contradictory sources.

SL is an example of a probabilistic approach, whereas in this paper we will outline a trust model that uses a gradual approach, meaning that agents can be trusted to some degree. Furthermore, to preserve provenance information, our model deals with distrust in addition to trust; hence, our intended approach is situated in the bottom right corner of Table 1. In this category, as far as we know, besides our earlier work [7,26], there is only one other existing model, namely Guha et al.'s [12]. They use a couple (t, d) with a trust degree t and a distrust degree d, both in [0,1]. To obtain the final suggested trust value, they subtract d from t. As we explain later on, potentially important information is lost when the trust and distrust scales are merged into one.

In the next section it will become clear that current gradual trust models are either not capable of properly handling inconsistency, or cannot differentiate unknown agents from malicious agents, although these problems can possibly have a large effect on (the ranking of) the recommendations. To deal with these issues, we introduce a new trust model which is able to solve "*trust problems*", caused by presence of distrust or lack of knowledge, and provides insight into "*knowledge problems*" caused by having too little or too much, i.e. contradictory, information.

3 A bilattice-based trust model

We propose an extension of [7] in which trust values are derived from a bilattice. Since their introduction by Ginsberg [10] in 1988, much attention has been paid to bilattices and their applications. It has e.g. been shown that bilattices are useful for providing semantics to logic programs (see e.g. [8]), and as underlying algebraic structures of formalisms for reasoning with imprecise information (see e.g. [6,9]). The use of these bilattices results in a new gradual model for (trust,distrust)-couples. We call such couples trust scores.

Definition 1 (Trust Score) *A trust score* (x_1, x_2) *is an element of* $[0, 1]^2$*, in which* x_1 *is called the trust degree, and* x_2 *the distrust degree.*

Trust scores will be used to compare the degree of trust and distrust an agent may have in other agents in the network. This information can be used in the ranking mechanisms of the RS, e.g. by giving preference to recommendations from sources that are trusted more. To this aim, we introduce the trust score space as a model that allows to compare and preserve information about the provenance of trust scores.

Definition 2 (Trust Score Space) *The trust score space*

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

consists of the set $[0, 1]^2$ *of trust scores, a trust ordering* $\leq_t$*, a knowledge ordering* $\leq_k$*, and a negation* $\neg$ *defined by*

$$\begin{aligned} (x_1, x_2) \leq_t (y_1, y_2) &\text{ iff } x_1 \leq y_1 \text{ and } x_2 \geq y_2 \\ (x_1, x_2) \leq_k (y_1, y_2) &\text{ iff } x_1 \leq y_1 \text{ and } x_2 \leq y_2 \\ \neg(x_1, x_2) &= (x_2, x_1) \end{aligned}$$

for all (x_1, x_2) *and* (y_1, y_2) *in* $[0, 1]^2$*.*

One can verify that the structure $\mathcal{BL}^{\square}$ is a bilattice in the sense of Ginsberg [10], that is $([0, 1]^2, \leq_t)$ and $([0, 1]^2, \leq_k)$ are both lattices and the negation $\neg$ serves to impose a relationship between them:

$$(x_1, x_2) \leq_t (y_1, y_2) \Rightarrow \neg(x_1, x_2) \geq_t \neg(y_1, y_2)$$

$$(x_1, x_2) \leq_k (y_1, y_2) \Rightarrow \neg(x_1, x_2) \leq_k \neg(y_1, y_2),$$

such that $\neg\neg(x_1, x_2) = (x_1, x_2)$. In other words, $\neg$ is an involution that reverses the $\leq_t$-order and preserves the $\leq_k$-order.

Fig. 4 shows $\mathcal{BL}^{\square}$, along with some examples of trust scores. These scores are interpreted as epistemic values: compared to Jøsang and Knapskog's subjective logic, the trust and distrust degrees are not complementary, but they reflect the imperfect knowledge we have about the actual trust and distrust values (which are complementary). The lattice $([0, 1]^2, \leq_t)$ 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

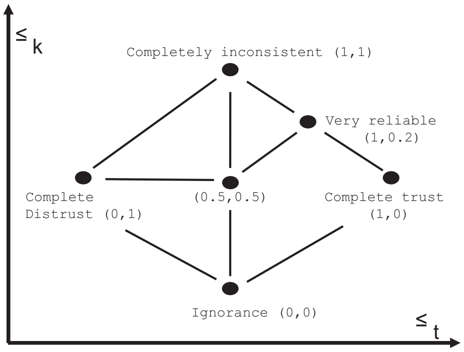

Fig. 4. Trust score space $\mathcal{BL}^{\square}$

"shortage of evidence", $x_1 + x_2 < 1$ (incomplete information), to an "excess of evidence", viz. $x_1 + x_2 > 1$ (inconsistent or contradictory information). In the extreme cases, there is no information available: $(0, 0)$; or there is evidence that says that the agent is to be trusted fully as well as evidence that states that the agent is completely unreliable: $(1, 1)$. Note that in this state, maximal knowledge occurs, while optimal knowledge occurs when the trust and distrust degree sum up to 1 (e.g. in the complete trust state).

The available trust information is modeled as a $\mathcal{BL}^{\square}$-fuzzy relation in the set of agents that associates a score drawn from the trust score space with each ordered pair of agents. It should be thought of as a snapshot taken at a certain moment since trust scores can be updated.

Definition 3 (Trust Network) *A trust network is a couple (A, R) such that A is a set of agents and R is an $A \times A \to [0, 1]^2$ mapping. For every a and b in A, we write*

$$R(a, b) = \left(R^+(a, b), R^-(a, b)\right)$$

- *$R(a, b)$ is called the trust score of a in b.*
- *$R^+(a, b)$ is called the trust degree of a in b.*
- *$R^-(a, b)$ is called the distrust degree of a in b.*

The following examples reveal some important shortcomings of current trust models which are alleviated by our bilattice model. Without harming generality, and to emphasize that trust is always dependent on a specific goal, task, or application, we focus on one kind of RSs, namely a movie recommender. The first example illustrates the need for models working with trust and distrust.

Example 1 (Ignorance without provenance) *Agent a wants to know if he should see a particular movie. Agents c and d have seen the movie, but a does not know them personally. So, in order to establish an opinion about c and d, a calls upon b for trust opinions on these agents. Agent b completely distrusts c when it comes to movies, hence b trusts c to degree 0 in the range [0,1], where 0 is full absence of trust and 1 full presence of trust. On the other* *hand b does not know d, hence b also trusts d to degree 0. As a result, b returns the same trust opinion to a for both c and d, namely 0, but the meaning of this value is clearly different in both cases.*

With *c*, the lack of trust is caused by a presence of distrust, while with *d*, the absence of trust is caused by a lack of knowledge. This provenance information is vital for *a* to make a well-informed decision. For example, if *a* has a high trust in TTP *b*, he will not consider *c* anymore, but might ask for other opinions on *d*. A trust model that takes into account both trust and distrust could be a possible solution. However, as the scenarios below illustrate, the existing approaches fall short because they do not allow to model inconsistency.

Example 2 (Contradictory information) *A stranger tells you that a particular movie was very bad. Because you do not know anything about this person, you make inquiries with two of your friends who are acquainted with him. One of them tells you to trust him, while the other tells you to distrust that same person. In this case, there are two equally trusted TTPs that tell you the exact opposite thing. In other words, you have to deal with inconsistent information.*

This example illustrates how inconsistencies may arise: when an agent in the trust network inquires for a trust estimation about another agent, it often happens that he does not ask one TTP's opinion, but several. Then these pieces of information, coming from different sources and propagated through different propagation chains, must be combined together into one new trust value which represents the opinion of all the TTP's. This is not an easy task when conflicting evidence has been gathered.

First of all, note that models that work only with trust and not with distrust are again not expressive enough to represent these cases adequately. Taking e.g. 0.5 (the average) as an aggregated trust value is not a good solution for Ex. 2, because then we cannot differentiate this case from the partial trust situation in which both of your friends trust the recommender to the extent 0.5, which indicates that the recommender is somewhat reliable. Furthermore, what would you answer if someone asks you if the stranger can be trusted? A plausible answer is: "I don't really know, because I have contradictory information about him". Note that this is fundamentally different from "I don't know, because I have no information about him". Hence, an aggregated trust value of 0 is not a suitable option either, as it could imply both inconsistency and ignorance.

In our bilattice model, these situations are respectively represented by $(1, 1)$ and $(0, 0)$. Note that previous models considering both trust and distrust degrees do not offer the option of representing (partial) inconsistency, even though this might arise quite naturally in large networks with contradictory sources. Jøsang's SL for example can not cope with this scenario because the belief and disbelief have to sum up to 1. A similar remark applies to our own previous work [7] in which we proposed to model the trust network as an intuitionistic fuzzy relation [4]. Guha et al. [12] do not impose a restriction on the trust and distrust degrees but their approach suffers from yet another kind of shortcoming, as the following example illustrates.

Example 3 (Ignorance without provenance) *Agent a needs to establish an opinion about agent c to find out whether he should follow c's recommendation to see a particular movie. Agent a may ask agent b for an opinion of c because agent a does not know anything about c. Agent b, in this case, is a TTP that knows how to compute a trust value of c from a web of trust. Assume that b has evidence for both trusting and distrusting c. For instance, let us say that b trusts c to degree 0.5 in the range [0,1] where 0 is full absence of trust and 1 is full presence of trust; and that b distrusts c to degree 0.2 in the range [0,1] where 0 is full absence of distrust and 1 is full presence of distrust. Another way of saying this is that b trusts c at least to the extent 0.5, but also not more than 0.8. The length of the interval [0.5,0.8] indicates how much information b lacks about c.*

Note once more that in this scenario, if we would only work with one value, by getting only the trust degree 0.5 from b, a is losing valuable information indicating that b has some evidence to distrust c too; this problem is solved by all models working with two values, and in particular Guha et al's. However, their approach has one main disadvantage: b will pass on a value of $0.5 - 0.2 = 0.3$ to a, i.e. the difference of the trust degree and the distrust degree, hence losing valuable trust provenance information indicating, for example, how much information b lacks about c. Note that a trust value of 0.3 and a distrust value of 0 also result in 0.3, but this scenario clearly differs from the former because now a has no reason to distrust b.

An important strength of our bilattice-based approach is that it can distinguish full distrust from ignorance. This is an improvement of e.g. [1,14,21,27]. Furthermore, it enables us to deal with both incomplete information and inconsistency, which is an improvement of [7] and [13]. Moreover, we do not lose important information because we keep the trust and distrust degree separated throughout the whole trust process (improvement of [12]). Whereas most of the other trust approaches assign a trust value of 0 to a malicious agent and consequently cannot differentiate a malicious agent from an unknown agent, in our approach a malicious agent gets a trust score of $(0, 1)$, while an unknown agent corresponds to a trust score of $(0, 0)$, leaving no room for confusion. Hence, if e.g. agents a, b, …, k all think that l is malicious, then the aggregated trust score for l will approach $(0, 1)$, indicating that l is regarded as a malicious insider. Hence, existing RS techniques may be made less vulnerable to recommendation attacks.

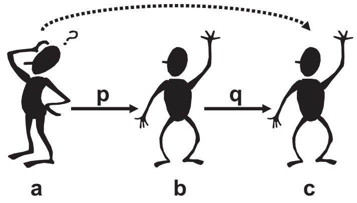

Fig. 5. Atomic propagation

4 Propagation of trust scores

In large networks, most of the other users are typically unknown to a specific user. Still there are cases in which it is useful to be able to derive some information on whether or not an unknown user can be trusted, and if so, to what degree. In the context of RSs this is important if none of the known agents has rated a specific item that the user is interested in, but there are some ratings available by unknown agents. In virtual trust networks, propagation operators are used to handle the problem of establishing trust information in an unknown agent by inquiring through TTPs. The simplest case, atomic propagation, is depicted in Fig. 5: if the trust score of agent a in agent b is p, and the trust score of b in agent c is q, what information can be derived about the trust score of a in agent c that is unknown to him? This is a reflection of real-life behaviour: to find out whether or not you should see a movie, you can ask your friend's opinion about it; if he has not seen the movie, he can ask a friend of his, and so on.

When dealing with trust only, multiplication is most often used as a propagation operator (see e.g. [11], [21] for a gradual, resp. probabilistic approach). In other words, when p and q in Fig. 5 are numbers in $[0, 1]$, the resulting trust value of a in c is $p \cdot q$. The underlying meaning is that a will trust c iff a trusts b and b trusts c. For reasons explained above, a model dealing with trust only, falls short, among other things, in distinguishing between a lack of trust caused by a lack of knowledge versus by a presence of distrust. Hence in this section we focus on the design of propagation operators in a model where p and q are elements of $[0, 1]^2$, i.e. couples of trust and distrust degrees, and the corresponding propagation operators are of type $[0, 1]^2 \times [0, 1]^2 \rightarrow [0, 1]^2$. We start the discussion with some intuitive examples.

Example 4 *If a friend whom you fully trust tells you to distrust someone, and you have no other information about this person, you likely will choose to distrust him. In other words, from $R(a, b) = (1, 0)$ and $R(b, c) = (0, 1)$ is derived that $R(a, c) = (0, 1)$, or, using $\mathcal{F}$ to denote an operator for trust score propagation:*

$$\mathcal{F}((1, 0), (0, 1)) = (0, 1) \tag{1}$$

Example 5 *If a colleague whom you distrust tells you to trust someone, you* *might decide this is too little information to act on. Indeed, if you distrust your colleague, it is reasonable not to take into account whatever he is telling you. Hence, from* $R(a,b) = (0,1)$ *and* $R(b,c) = (1,0)$ *is derived that* $R(a,c) = (0,0)$*. This means that, if no additional information is available, c remains unknown to you. Using* $\mathcal{F}$ *to denote an operator for trust score propagation, this comes down to*

$$\mathcal{F}((0,1),(1,0)) = (0,0) \tag{2}$$

On the other hand, you might think that the colleague you distrust is giving you wrong information on purpose, or you might conclude that trusted friends of your distrusted colleague are best also to be distrusted. In this case, from $R(a,b) = (0,1)$ *and* $R(b,c) = (1,0)$ *is derived that* $R(a,c) = (0,1)$*, hence we are confronted with a propagation operator* $\mathcal{G}$ *which is clearly different from* $\mathcal{F}$ *since*

$$\mathcal{G}((0,1),(1,0)) = (0,1) \tag{3}$$

The examples above serve two purposes: first, (1) and (2) illustrate that a trust score propagation operator is not necessarily commutative because the ordering of the arguments matters. Note how this is different from the traditional case where the commutative multiplication seems sufficient to do the job. Secondly, (2) and (3) illustrate that different operators yielding different results are possible depending on the interpretation, thus revealing part of the complex problem of choosing an appropriate propagation scheme for the application at hand. Our aim in this paper is not to provide a clear cut answer to that question, but rather to provide some propagation operators that can be used in different schemes, as well as to discuss some of their properties.

We call a propagation operator $\mathcal{F}$ knowledge monotonic if the arguments can be replaced by higher trust scores w.r.t. the knowledge ordering $\leq_k$ without decreasing the resulting trust score. Knowledge monotonicity reflects that the better agent a knows agent b with whom it is inquiring about agent c, the more informed a will be about how well to trust or distrust agent c.

Definition 4 (Knowledge Monotonicity) *A propagation operator* $\mathcal{F}$ *on* $[0,1]^2$ *is said to be knowledge monotonic iff for all* x*,* y*,* z*, and* u *in* $[0,1]^2$*,*

$$x \leq_k y \text{ and } z \leq_k u \text{ implies } \mathcal{F}(x,z) \leq_k \mathcal{F}(y,u)$$

Knowledge monotonicity is not only useful to provide more insight in the propagation operators but it can also be used to establish a lower bound w.r.t. $\leq_k$ for the actual propagated trust score without immediate recalculation. This might be useful in a situation where one of the agents has gained more knowledge about another agent and there is not enough time to recalculate the whole propagation chain immediately, as will be the case for many RS types. The analogue for the trust ordering $\leq_t$ is not a useful property, because it counteracts normal real-life behaviour as we illustrate next.

Example 6 *If a new colleague tells you to distrust someone, you might decide not to take into account his opinion because you do not know him sufficiently. Using $\mathcal{F}$ to denote an operator for trust propagation, this comes down to*

$$\mathcal{F}((0,0),(0,1)) = (0,0) \tag{4}$$

However, over time this colleague might become a trusted friend, i.e. your trust in your colleague increases, and you will start distrusting others because your colleague tells you to (see (1)). In this case the trust score in one of the links of the chain goes up from (0,0) to (1,0) while the overall trust score of the chain drops from (0,0) to (0,1).

Besides atomic propagation, we need to be able to consider longer propagation chains, so TTPs can in turn consult their own TTPs and so on. For an associative propagation operator, this extension can be defined unambiguously. In particular, with an associative propagation operator, the overall trust score computed from a longer propagation chain is independent of the choice of which two subsequent trust scores to combine first. Even for non associative propagation operators, instead of calculating a whole chain, sometimes it is sufficient to look at only one agent to determine the overall trust score in a longer propagation chain.

Definition 5 (Knowledge absorption) *A propagation operator $\mathcal{F}$ on $[0,1]^2$ is said to be knowledge absorbing iff for all x and y in $[0,1]^2$,*

$$\mathcal{F}((0,0),y) = \mathcal{F}(x,(0,0)) = (0,0)$$

Hence, as soon as one of the agents is ignorant, we can dismiss the entire chain. As such, for an operator with this property, precious calculation time can possibly be saved.

The propagation operators we will define below are constructed using the fuzzy logical operators for conjunction, disjunction and negation. 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 $\mathcal{N}(1) = 0$. In the remainder of this section, we use t_1 as an abbreviation for the trust degree $R^+(a,b)$ of agent a in agent b, and d_1 for the corresponding distrust degree $R^-(a,b)$. Similarly, we use (t_2,d_2) to denote the trust score from agent b in agent c. In other words $R(a,b) = (t_1,d_1)$ and $R(b,c) = (t_2,d_2)$.

People are likely to listen to whom they trust; this attitude is reflected by the first propagation operator.

Definition 6 *The propagation operator* $\texttt{Prop}_1$ *is defined by*

$$\texttt{Prop}_1((t_1, d_1), (t_2, d_2)) = (\mathcal{T}(t_1, t_2), \mathcal{T}(t_1, d_2))$$

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

An agent with this profile ($\texttt{Prop}_1$) exhibits a skeptical behaviour in deriving no knowledge through a distrusted or unknown third party. In the upper left corner of Table 2, the behaviour of $\texttt{Prop}_1$ for binary inputs is shown. Note that the results for inconsistency in the last link are also in accordance with this behaviour. We do not consider results for inconsistency in the first link, because we assume that all agents behave in a consistent way; in fact, it is only useful to propagate inconsistency when it occurs in the last link of the propagation chain (where information is possibly aggregated).

It follows from the monotonicity of $\mathcal{T}$ that $\texttt{Prop}_1$ is knowledge monotonic, while associativity of the t-norm leads to $\texttt{Prop}_1$ being associative. If there occurs a "missing link" $(0, 0)$ anywhere in the propagation chain, the result will contain no useful information. In other words, the propagation operator is knowledge absorbing. Note that the same conclusion (i.e. ignorance) can be drawn if at any position in the chain, except the last one, there occurs complete distrust $(0, 1)$.

Using the product t-norm $\mathcal{T}_P$, defined as $\mathcal{T}_P(x, y) = x \cdot y$, $\texttt{Prop}_1$ takes on the following form

$$\texttt{Prop}_1((t_1, d_1), (t_2, d_2)) = (t_1 \cdot t_2, t_1 \cdot d_2)$$

This particular form of $\texttt{Prop}_1$ has previously been proposed in [13] to combine pairs of beliefs and disbeliefs. Subtracting the distrust degree from the trust degree, this propagation operator reduces to $t_1 \cdot (t_2 - d_2)$, a propagation scheme proposed in [12].

$\texttt{Prop}_1$ neglects all information coming from an unknown agent. However, some agents might be willing to take over some information coming from whatever party, as long as it is not distrusted. For instance, when agent b warns a about an agent c that is to be distrusted, agent a might listen to the advice even when he does not know b. In this way we arrive at a propagation operator reflecting that a trusts c when a trusts b *and* b trusts c (in other words, the classical behaviour), and a distrusts c because b distrusts c *and* a does *not* distrust b.

Definition 7 *The propagation operator* $\texttt{Prop}_2$ *is defined by*

$$\texttt{Prop}_2((t_1, d_1), (t_2, d_2)) = (\mathcal{T}(t_1, t_2), \mathcal{T}(\mathcal{N}(d_1), d_2))$$

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

Table 2
Propagation operators, using TTP b with $R(a,b) = (t_1, d_1)$ (rows) and $R(b,c) = (t_2, d_2)$ (columns)

$\texttt{Prop}_1$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$
$(0,1)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$
$(1,0)$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$

$\texttt{Prop}_2$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,0)$	$(0,0)$	$(0,1)$	$(0,0)$	$(0,1)$
$(0,1)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$
$(1,0)$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$

$\texttt{Prop}_3$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$
$(0,1)$	$(0,0)$	$(1,0)$	$(0,1)$	$(1,1)$
$(1,0)$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$

$\texttt{Prop}_4$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$
$(0,1)$	$(0,0)$	$(0,0)$	$(0,1)$	$(0,1)$
$(1,0)$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$

Note how $\texttt{Prop}_2$ only differs from $\texttt{Prop}_1$ in the computation of the propagated distrust degree. Agents with this second profile ($\texttt{Prop}_2$) display a paranoid behaviour in taking some distrust information even from an unknown third party: suppose you meet someone that tells you that movie m was dreadful. Even though you do not know this person and whether to trust him, it may happen that you retain some of this negative information. This paranoid behaviour also occurs when the unknown third party receives inconsistent information. The following example illustrates that $\texttt{Prop}_2$ is not knowledge monotonic.

Example 7 *In this example we use the standard negator $\mathcal{N}_s$, defined by $\mathcal{N}_s(x) = 1 - x$, and an arbitrary t-norm. To see that $\texttt{Prop}_2$ is not knowledge monotonic, consider*

$$\texttt{Prop}_2((0.2, 0.7), (0, 1)) = (0, 0.3)$$
$$\texttt{Prop}_2((0.2, 0.8), (0, 1)) = (0, 0.2)$$

Going from the first to the second situation, all trust degrees remain the same but the distrust degree of agent a in agent b has increased slightly. In other words, a has formed a slightly more informed opinion about b:

$$(0.2, 0.7) \leq_k (0.2, 0.8)$$

and trivially also $(0,1) \leq_k (0,1)$. However at the same time the distrust degree of a in c has dropped slightly; since the trust degree of a in c did not change, agent a now has slightly less knowledge about c:

$$(0, 0.3) \not\leq_k (0, 0.2)$$

The intuitive explanation behind the non knowledge monotonic behaviour of

$\texttt{Prop}_2$ is that, using this propagation operator, agent a takes over distrust from a stranger b, hence giving b the benefit of the doubt, but when a starts to distrust b (thus knowing b better), a will adopt b's opinion to a lesser extent, or in other words, derive less knowledge. $\texttt{Prop}_2$ is knowledge absorbing but not associative as the following example shows.

Example 8 *Using the standard negator $\mathcal{N}_s$ and the product t-norm $\mathcal{T}_P$ we obtain:*

$$\begin{aligned}\texttt{Prop}_2((0.3,0.6),\texttt{Prop}_2((0.1,0.2),(0.8,0.1))) &= (0.024,0.032)\\ \texttt{Prop}_2(\texttt{Prop}_2((0.3,0.6),(0.1,0.2)),(0.8,0.1)) &= (0.024,0.092)\end{aligned}$$

The following example illustrates the effects of gradual trust and gradual distrust.

Example 9 *In this example we use the product t-norm $\mathcal{T}_P$ and the standard negator $\mathcal{N}_s$. Assume that, although agent a highly trusts b, there is also evidence to slightly distrust b, e.g.*

$$(t_1,d_1)=(0.8,0.2)$$

Furthermore assume that b highly distrusts c, i.e.

$$(t_2,d_2)=(0.1,0.9)$$

Then, if agent a matches the second profile, we obtain

$$\texttt{Prop}_2((t_1,d_1),(t_2,d_2))=(0.08,0.72)$$

In other words, agent a takes over most of the information that b provides; however, the final trust score is mitigated because a also slightly distrusts b.

Unlike the first profiles, it is in fact possible that some agents will use information coming from a distrusted agent, as is the case in (3). Propagation operator $\texttt{Prop}_3$ is an extension of $\texttt{Prop}_1$ in which agent a assumes that a distrusted agent b gives the wrong information on purpose. Hence, instead of simply ignoring a distrusted agent b, agent a assumes the opposite of what b is telling him. To achieve this, the trust and distrust degrees are computed as disjunctions, modelled by means of a t-conorm $\mathcal{S}$. For example, as in the first profile, agent a will distrust c when a trusted agent b will tell him to, but in addition, a will also distrust c when there is an advice to trust c coming from a distrusted agent b.

Definition 8 *The propagation operator* $\texttt{Prop}_3$ *is defined by*

$$\texttt{Prop}_3((t_1,d_1),(t_2,d_2))=(\mathcal{S}(\mathcal{T}(t_1,t_2),\mathcal{T}(d_1,d_2)),\mathcal{S}(\mathcal{T}(t_1,d_2),\mathcal{T}(d_1,t_2)))$$

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

Agents that fit this profile consider an enemy of an enemy to be a friend:

$$\texttt{Prop}_3((0,1),(0,1)) = (1,0)$$

and a friend of an enemy to be an enemy:

$$\texttt{Prop}_3((0,1),(1,0)) = (0,1)$$

with friend (enemy) denoting a person that is (dis)trusted. Due to the monotonicity of $\mathcal{T}$ and $\mathcal{S}$, $\texttt{Prop}_3$ is knowledge monotonic. Examples can be constructed to prove that $\texttt{Prop}_3$ is not associative. Knowledge absorption holds for $\texttt{Prop}_3$, despite the fact that it is not associative.

Using t-norm $\mathcal{T}_P$ and the corresponding t-conorm $\mathcal{S}_P$, the operator becomes

$$\texttt{Prop}_3((t_1,d_1),(t_2,d_2)) = (t_1 \cdot t_2 + d_1 \cdot d_2 - t_1 \cdot t_2 \cdot d_1 \cdot d_2, t_1 \cdot d_2 + d_1 \cdot t_2 - t_1 \cdot d_2 \cdot d_1 \cdot t_2)$$

Subtracting the distrust degree from the trust degree, we obtain $(t_1 - d_1) \cdot (t_2 - d_2)$, the distrust propagation scheme put forward in [12].

The last profile ($\texttt{Prop}_4$) is a moderation of $\texttt{Prop}_3$ in such a way that these agents do not take over information coming from a distrusted agent c when b is distrusted. While a friend of an enemy is still considered to be an enemy, no information is derived about an enemy of an enemy:

$$\texttt{Prop}_4((0,1),(0,1)) = (0,0)$$

$$\texttt{Prop}_4((0,1),(1,0)) = (0,1)$$

In other words, c is only trusted by a when a trusts b and b trusts c. The properties of the third profile apply to $\texttt{Prop}_4$ as well.

Definition 9 *The propagation operator* $\texttt{Prop}_4$ *is defined by*

$$\texttt{Prop}_4((t_1,d_1),(t_2,d_2)) = (\mathcal{T}(t_1,t_2), \mathcal{S}(\mathcal{T}(t_1,d_2), \mathcal{T}(d_1,t_2)))$$

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

In summary, as Proposition 1 shows, all the proposed propagation operators copy information from a fully trusted TTP. All of them ignore information coming from an unknown party, except $\texttt{Prop}_2$ which takes over the distrust information from a stranger.

Proposition 1 *For all* (t,d) *in* $[0,1]^2$ *it holds that*

(1) $\texttt{Prop}_1((1,0),(t,d)) = (t,d)$
(2) $\texttt{Prop}_2((1,0),(t,d)) = (t,d)$

(3) $\texttt{Prop}_3((1,0),(t,d)) = (t,d)$
(4) $\texttt{Prop}_4((1,0),(t,d)) = (t,d)$
(5) $\texttt{Prop}_1((0,0),(t,d)) = (0,0)$
(6) $\texttt{Prop}_2((0,0),(t,d)) = (0,d)$
(7) $\texttt{Prop}_3((0,0),(t,d)) = (0,0)$
(8) $\texttt{Prop}_4((0,0),(t,d)) = (0,0)$
(9) $\texttt{Prop}_1((0,1),(t,d)) = (0,0)$
(10) $\texttt{Prop}_2((0,1),(t,d)) = (0,0)$
(11) $\texttt{Prop}_3((0,1),(t,d)) = (d,t)$
(12) $\texttt{Prop}_4((0,1),(t,d)) = (0,t)$

Furthermore, $\texttt{Prop}_3$ and $\texttt{Prop}_4$ allow to derive useful distrust information in the context of RSs, as the following examples illustrate.

Example 10 *Suppose that agent a fully distrusts agent b, and b fully distrusts c. Trust-enhanced RSs working with only one value will not use this information. However, in our approach, e.g., if c rates m highly and agent a fits the third profile, movie m might be a good recommendation for a.*

In this way, these profiles may further help us to alleviate the sparsity and cold start problem; moreover, they can also be used to filter out false positives generated by other RS techniques, as illustrated by the next example.

Example 11 *Suppose that movie m is highly rated by recommender c and that b trusts c, while a completely distrusts b. If a fits the third or fourth profile, m could serve as a negative recommendation for a. Hence, m should be filtered out if it appears in the list generated by the CF/CB approach.*

To illustrate that the different propagation behaviours discussed above appear in real life trust networks, Table 3 contains relevant statistical information on a dataset from Epinions. The Epinions' web of trust graph from the dataset contains 131 829 users who issued 840 799 trust or distrust statements. About 85% of them are labelled as trust. If user X included user Y in his web of trust, we assume a trust score $(1,0)$ of X in Y. Similarly, the fact that X included Y in his block list, results in a trust score $(0,1)$ of X in Y. Finally, if two users in the dataset are not linked by a trust or distrust statement, we interpret this as a trust score $(0,0)$, modeling ignorance. Inconsistency links do not exist in the dataset because users cannot put someone in their web of trust and block list simultaneously. To investigate the behaviour reflected by the propagation operators, we focus on triples (X,Y,Z) in which the trust score of X in Z is either $(1,0)$ or $(0,1)$. We do not include chains in which $X-Z$ represents ignorance because the portion of triples in which $X-Z$ represents a trust or distrust relation, over the triples with no restriction on $X-Z$, is very small. This means that most of these chains would propagate to $(0,0)$, which gives a distorted picture of the real, useful, patterns in the

Table 3
Evaluation propagation profiles

$X-Y$	$Y-Z$	$X-Z$	# (X,Y,Z)
(1,0)	(1,0)	(1,0)	9 594 233
(1,0)	(0,1)	(0,1)	271 534
(0,1)	(1,0)	(0,1)	296 078
(0,1)	(0,1)	(1,0)	52 385
(0,0)	(0,1)	(0,1)	3 606 754

dataset.

Results can be found in Table 3. The first column denotes the relation between X and Y, the second and third column between Y and Z, and X and Z respectively. The fourth column contains the number of chains that consist of the given $X-Y$, $Y-Z$ and $X-Z$ link. The first row shows us that the behaviour of trusting an agent which is trusted by a TTP is omnipresent; this is the attitude which is at the basis of all proposed propagation operators. The second, third and fourth row illustrate that distrust can be used in the propagation process in several ways. The second row reflects the behaviour that can be found in all our propagation operators, viz. distrusting agents who are distrusted by a TTP. The third and fourth row exhibit the behaviour of $\texttt{Prop}_3$ and $\texttt{Prop}_4$, viz. a friend of an enemy is an enemy, and an enemy of an enemy is a friend. Evidence in favor of $\texttt{Prop}_2$ can be found by focussing on ignorance-distrust chains which retain the distrust information (fifth row).

These results show us that the proposed propagation behaviours occur in the dataset. Note that the number of chains in the first row of Table 3 is significantly higher than the rest. This is due to the fact that the other rows contain at least one distrust link, while the number of available distrust links in the dataset is small overall[5]. Taking this into account, the other results (reflecting behaviour of $\texttt{Prop}_2$, $\texttt{Prop}_3$ and $\texttt{Prop}_4$) cannot/may not be ignored because they clearly indicate that distrust can be used in the propagation process, via the first as well as the second link. These results illustrate one of the main benefits of our approach: because the bilattice-based model can properly represent distrust information, we are able to include it in the recommendation process, while other trust-based algorithms cannot.

[5] In the last row of Table 3, this phenomenon is partially compensated by the abundant number of ignorance links, caused by the sparsity of the web of trust graph.

5 Concluding remarks

The incorporation of a trust network among the users of a RS proves beneficial to the quality and amount of recommendations, thereby alleviating important problems of collaborative filtering like cold start and sparsity. However, as current trust-enhanced RSs do not preserve vital trust provenance information, they cannot cope with "*trust problems*" caused by presence of distrust versus by lack of knowledge, and "*knowledge problems*" caused by having too little or too much, i.e. contradictory, information.

To resolve these issues, we represent trust scores as elements (t, d) of the bilattice $\mathcal{BL}^{\square} = ([0,1]^2, \leq_t, \leq_k, \neg)$ in which t corresponds to a trust degree and d to a distrust degree. As such, to our knowledge, we are the first to introduce a model that takes into account gradual trust, distrust, ignorance and inconsistency simultaneously. We have also presented a collection of four operators for atomic propagation that are generic enough to be used in several trust schemes, including those where trust and distrust are either binary or gradual. The ability to handle ignorance and inconsistency and to propagate trust becomes extremely meaningful in a large web of RS users where the trustworthiness of many agents is initially unknown to a user, which does not imply that the user distrusts all of them, but that the user may eventually gather evidence to trust or distrust some agents and still ignore others.

Apart from the propagation solutions presented in this paper, there are a number of other issues that need to be addressed, such as further propagation (longer chains) and aggregation. On another count, existing trust techniques for RSs require a central authority to propagate and aggregate trust values; however, as the amount of customers of RSs continues to grow, it will get more and more difficult to manage all trust information in one place, so a decentralized approach may be more appropriate. Furthermore, privacy of data is becoming increasingly important in web applications, and RS users may refuse to disclose their personal web of trust. A key problem therefore is how to best combine the available trust scores and recommendations, a decision that will impact the responsibility and autonomy of the agents in the network.

Acknowledgements

Patricia Victor would like to thank the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) for funding her research. Chris Cornelis would like to thank the Research Foundation-Flanders for funding his research.

The authors would like to thank Epinions.com for making the data available, and in particular R. Guha, R. Kumar, P. Raghavan and A. Tomkins.

References

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

[2] F. Almenárez, A. Marín, C. Campo, and C. García, PTM: A pervasive trust management model for dynamic open environments, First Workshop on Pervasive Security, Privacy and Trust, PSPT2004 in conjuntion with Mobiquitous 2004, 2004.

[3] D. Artz and Y. Gil, A survey of trust in computer science and the semantic web, Journal of Web Semantics 5, pages 58–71, 2007.

[4] K.T. Atanassov, Intuitionistic fuzzy sets, Fuzzy Sets and Systems 20, pages 87–96, 1986.

[5] T. Berners-Lee, J. Hendler, and O. Lassila, The Semantic Web, Scientific American May 2001, 35–43, 2001.

[6] C. Cornelis, O. Arieli, G. Deschrijver, and E.E. Kerre, Uncertainty modeling by bilattice-based squares and triangles, IEEE T Fuzz Syst 15, pages 161–175, 2007.

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

[8] M. Fitting, Bilattices and the semantics of logic programming, Journal of Logic Programming 11, pages 91–116, 1991.

[9] G. Gargov, Knowledge, uncertainty and ignorance in logic: bilattices and beyond, Journal of Applied Non-Classical Logics 9, pages 195–283, 1999.

[10] M. Ginsberg, Multi-valued logics: A uniform approach to reasoning in artificial intelligence, Comput Intell 4, pages 265–316, 1988.

[11] J. Golbeck, Computing and applying trust in web-based social networks, PhD thesis, 2005.

[12] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, Propagation of trust and distrust, Proceedings of WWW2004, pages 403–412, 2004.

[13] A. Jøsang and S. Knapskog, A metric for trusted systems, Proceedings of NIST-NCSC 1998, pages 16–29, 1998.

[14] S. Kamvar, M. Schlosser, and H. Garcia-Molina, The eigentrust algorithm for reputation management in P2P networks, Proceedings of WWW2003, pages 640–651, 2003.

[15] P. Massa and P. Avesani, Trust-aware collaborative filtering for recommender systems, Proceedings of the Federated International Conference On The Move to Meaningful Internet: CoopIS, DOA, ODBASE, pages 492–508, 2004.

[16] J. O'Donovan and B. Smyth, Trust in Recommender Systems, Proceedings of the 10th international conference on Intelligent user interfaces, pages 167–174, 2005.

[17] R. Falcone, G. Pezzulo, and C. Castelfranchi, A fuzzy approach to a belief-based trust computation, Lecture Notes in Artificial Intelligence 2631, pages 73–86, 2003.

[18] T. O'Reilly, What is web 2.0, 2005. Available at `http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html`.

[19] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl, Grouplens: An open architecture for collaborative filtering of netnews, Proceedings of CSCW 1994, pages 175–186, 1994.

[20] P. Resnick and H.R. Varian, Recommender Systems. Communications of the ACM 40(3), pages 56–58, 1997.

[21] M. Richardson, R. Agrawal, and P. Domingos, Trust management for the semantic web, Proceedings of the Second International Semantic Web Conference, pages 351–368, 2003.

[22] J. Schafer, J. Konstan, and J. Riedl, E-commerce recommendation applications, Data Min Knowl Disc 5, pages 115–153, 2001.

[23] R. Sinha and K. Swearingen, Comparing recommendations made by online systems and friends, Proceedings of the DELOS-NSF Workshop on Personalisation and Recommender Systems in Digital Libraries, 2001.

[24] K. Stølen, W.H. Winsborough, F. Martinelli and F. Massacci (Eds.), Trust Management: 4th International Conference iTrust 2006, Lecture Notes in Computer Science 3986, 2006.

[25] W. Tang, Y. Ma, and Z. Chen, Managing trust in peer-to-peer networks, Journal of Digital Information Management 3, pages 58–63, 2005.

[26] P. Victor, M. De Cock, C. Cornelis, and P. Pinheiro da Silva, Towards a provenance-preserving trust model in agent networks, Proceedings of Models of Trust for the Web WWW2006 Workshop, 2006.

[27] I. Zaihrayeu, P. Pinheiro da Silva, and D. McGuinness, IWTrust: Improving user trust in answers from the web, Proceedings of the Third International Conference On Trust Management, pages 384–392, 2005.

[28] S. Zhang, Y. Ouyang, J. Ford, and F. Makedon, Analysis of a Low-Dimensional Linear Model Under Recommendation Attacks, Proceedings of ACM SIGIR 2006, pages 517–524, 2006.

[29] C. Ziegler, Semantic web recommender systems, Lecture Notes in Computer Science 3268, pages 78–89, 2004.