Chapter 20
Trust and Recommendations

Patricia Victor, Martine De Cock, and Chris Cornelis

Abstract Recommendation technologies and trust metrics constitute the two pillars of trust-enhanced recommender systems. We discuss and illustrate the basic trust concepts such as trust and distrust modeling, propagation and aggregation. These concepts are needed to fully grasp the rationale behind the trust-enhanced recommender techniques that are discussed in the central part of the chapter, which focuses on the application of trust metrics and their operators in recommender systems. We explain the benefits of using trust in recommender algorithms and give an overview of state-of-the-art approaches for trust-enhanced recommender systems. Furthermore, we explain the details of three well-known trust-based systems and provide a comparative analysis of their performance. We conclude with a discussion of some recent developments and open challenges, such as visualizing trust relationships in a recommender system, alleviating the cold start problem in a trust network of a recommender system, studying the effect of involving distrust in the recommendation process, and investigating the potential of other types of social relationships.

20.1 Introduction

Collaboration, interaction and information sharing are the main driving forces of the current generation of web applications referred to as 'Web 2.0' [48]. 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

Patricia Victor · Chris Cornelis
Dept.of Applied Mathematics and Computer Science, Ghent University, Krijgslaan 281 (S9), 9000 Gent, Belgium e-mail: Patricia.Victor,Chris.Cornelis@ugent.be

Martine De Cock
Institute of Technology, University of Washington Tacoma, 1900 Pacific Ave, Tacoma, WA, USA (on leave from Ghent University) e-mail: mdecock@u.washington.edu

[1] See www.foaf-project.org

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 Facebook[3], dating sites, car addict forums, etc.). In this chapter, we focus on one specific set of Web 2.0 applications, namely *social recommender systems*. These recommender systems generate predictions (recommendations) that are based on information about users' profiles and relationships between users. Nowadays, such online relationships can be found virtually everywhere, think for instance of the very popular social networking sites Facebook, LinkedIn and MSN[4].

Research has pointed out that people tend to rely more on recommendations from people they trust (friends) than on online recommender systems which generate recommendations based on anonymous people similar to them [57]. This observation, combined with the growing popularity of open social networks and the trend to integrate e-commerce applications with recommender systems, has generated a rising interest in *trust-enhanced recommendation systems*. The recommendations generated by these systems are based on information coming from an (online) *trust network*, a social network which expresses how much the members of the community trust each other. A typical example is Golbeck's FilmTrust [16], an online social network combined with a movie rating and review system in which users are asked to evaluate their acquaintances' movie tastes on a scale from 1 to 10. Another example is the e-commerce site Epinions.com, which maintains a trust network by asking its users to indicate which members they trust (i.e., their personal 'web of trust') or distrust ('block list').

Trust-enhanced recommender systems use the knowledge that originates from such trust networks to generate more personalized recommendations: users receive recommendations for items rated highly by people in their web of trust (WOT), or even by people who are trusted by these WOT members, etc. (see e.g. [7, 16, 46, 61]). The main strength of most of these systems is their use of *trust propagation* and *trust aggregation* operators; mechanisms to estimate the trust transitively by computing how much trust a user a has in another user c, given the value of trust for a trusted third party b by a and c by b (propagation), and by combining several trust estimates into one final trust value (aggregation). Propagation and aggregation are the two key building blocks of *trust metrics*, which aim to estimate the trust between two unknown users in the network.

Apart from trust, in a large group of users (each with their own intentions, tastes and opinions) it is only natural that also *distrust* occurs. For example, Epinions first provided the possiblity to include users in a personal WOT (based on their quality as a reviewer), and later on also introduced the concept of a personal 'block list', reflecting the members that are distrusted by a particular user. The information in

[2] See www.wikipedia.org

[3] See www.facebook.com

[4] See www.linkedin.com, or www.msn.com

the WOT and block list is then used to make the ordered list of presented reviews more personalized. From a research perspective, too, it is generally acknowledged that distrust can play an important role [21, 62, 68], but much ground remains to be covered in this domain.

Recommendation technologies and trust metrics constitute the two pillars of trust-enhanced recommender systems. Since the former are covered in much detail in other chapters of this handbook, we will restrict ourselves to the essentials. On the other hand, we do not assume that most readers are familiar with the trust research area. Therefore, in the following section, we start with a discussion and illustration of the basic trust concepts, namely trust and distrust modeling, propagation and aggregation; concepts that are needed to fully grasp the rationale behind the trust-enhanced recommender techniques as they are discussed in Section 3. This is the central part of the chapter, and focuses on the application of trust metrics and their operators in recommender systems. We explain the benefits of using trust in recommender algorithms and give an overview of state-of-the-art approaches for trust-enhanced recommender systems. Furthermore, we explain the details of three well-known trust-based systems and provide a comparative analysis of their performance. After this overview of classical trust-enhanced research, in Section 4, we focus on some recent developments and open challenges, such as visualizing trust relationships in a recommender system, alleviating the cold start problem in a trust network of a recommender system, studying the effect of involving distrust in the recommendation process, and investigating the potential of other types of social relationships. The chapter is concluded in Section 5.

20.2 Computational Trust

In this section we provide the reader with a basic introduction to the field of computational interpersonal trust, i.e., trust that can be computed among two individuals in a social trust network. This implies that we cover trust models (how to represent trust and how to deal with distrust; Section 2.1), trust propagation operators (how to estimate the trust between two individuals by using information coming from users that are on the connecting path between them; Section 2.2.1), and trust aggregation (how to combine trust values generated by multiple propagation paths; Section 2.2.2). We illustrate these concepts by classical and recent examples.

Note that this overview is inexhaustive; for instance, we do not cover trust updating or trust bootstrapping. Our primary goal is to familiarize the reader with the main concepts of the trust computation area, and as such to lay the foundation for an easy understanding of the rationale and details of the trust-enhanced recommendation techniques presented in Section 3.

20.2.1 Trust Representation

Trust models come in many flavours and can be classified in several ways. In this chapter we focus on two such classifications, namely probabilistic versus gradual approaches, and representations of trust versus representations of both trust and distrust. Table 20.1 shows 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. In such a setting, a higher suggested trust value corresponds to a higher probability that an agent can be trusted. Examples can, among others, be found in [66] in which Zaihrayeu et al. present an extension of an inference infrastructure that takes into account the trust between users and between users and provenance elements in the system, in [55] where the focus is on computing trust for applications containing semantic information such as a bibliography server, or in contributions like [32] in which a trust system is designed to make community blogs more attack-resistant. Trust is also often based on the number of positive and negative transactions between agents in a virtual network, such as in Kamvar et al.'s Eigentrust for peer-to-peer (P2P) networks [28], or Noh's formal model based on feedbacks in a social network [44]. Both [25] and [51] use a subjective logic framework (discussed later on in this section) to represent trust values; the former for quantifying and reasoning about trust in IT equipment, and the latter for determining the trustworthiness of agents in a P2P system.

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, 11, 15, 21, 35, 59, 68]). In a gradual setting, trust values are not interpreted as probabilities: a higher trust value corresponds to a higher trust in an agent, which makes the ordering of trust values a very important factor in such scenarios. 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 [29, 65] is very well-suited to represent such natural language labels which represent vague intervals rather than exact values. For instance, in [59] and [31], fuzzy linguistic terms are used to specify the trust in agents in a P2P network, and in a social network, respectively. A classical example of trust as a gradual notion can be found in [1], in which a four-value scale is used to determine the trustworthiness of agents, viz. very trustworthy - trustworthy - untrustworthy - very untrustworthy.

The last years have witnessed a rapid increase of gradual trust approaches, ranging from socio-cognitive models (for example implemented by fuzzy cognitive maps in [12]), over management mechanisms for selecting good interaction partners on the web [59] or for pervasive computing environments (Almenárez et al.'s PTM [3]), to representations for use in recommender systems [15, 35], and general models tailored to semantic web applications [68].

Table 20.1: Classification of trust models

	trust only	*trust and distrust*
probabilistic	Kamvar et al. [28] Richardson et al. [55] Zaihrayeu et al. [66]	Jøsang et al. [25]
gradual	Abdul-Rahman et al. [1] Falcone et al. [12] Golbeck [15] Massa et al. [35]	Victor et al. [62] Guha et al. [21]

While trust is increasingly getting established, the use and modeling of *distrust* remains relatively unexplored. Most approaches completely ignore distrust (see for example [31, 32, 43, 55, 66]), or consider trust and distrust as opposite ends of the same continuous scale (see e.g. [1, 19, 59]). However, in agent network theory there is a growing body of opinion that distrust cannot be seen as the equivalent of lack of trust [10, 13, 34]. Moreover, work in the psychology area has repeatedly asked for a re-examination of the assumption that positive- and negative-valent feelings are not separable [8, 50, 52], and some researchers even claim that trust and distrust are not opposite, but related dimensions that can occur simultaneously [9, 33].

To the best of our knowledge, there is only one probabilistic model that considers trust and distrust simultaneously: in Jøsang's subjective logic [24, 25], 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, but 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 [60].

Examples of gradual models for both trust and distrust can be found in [11, 21, 62, 68]. Guha et al. 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 [21]. However, as explained in [62], potentially important information is lost when the trust and distrust scales are merged into one. For example, the scenario $(0.2,0)$ in which there is partial trust collapses to 0.2, but so does the scenario $(0.6,0.4)$ that exhibits both partial trust *and* partial distrust. To deal with the issues in Guha's and Jøsang's approach, Victor et al. proposed an extension of [11] in which trust and distrust values are drawn from a bilattice [14]. Such a bilattice structure 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 [62].

Trust and trust models have been used in many fields of computer science, and also in a wide range of applications; a nice overview can be found in [6] in which

Artz and Gil classify trust research in four major areas: models that use policies to establish trust (enforcing access policies, managing credentials, etc.), general trust models such as [12] and [68], models for trust in information sources such as [66], and reputation-based trust models. The latter category includes, among others, research that uses the history of an agent's actions or behaviour (see e.g. [28, 46]), and work that computes trust over social networks, such as [21, 36]. In fact, the trust-enhanced recommender techniques that we will describe in Section 20.3 all belong to this class.

20.2.2 Trust Computation

In online trust networks, most 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 recommender systems for instance, this is important if none of the known users has rated a specific item that the user is interested in, but there are some ratings available by unknown users (who are a member of the trust network). For instance, the number of people that users have in their web of trust in Epinions is estimated to be around 1.7 on average. The total number of users of Epinions on the other hand well exceeds 700 000 [61]. In other words, the WOT of a user only contains a very tiny fraction of the user community. Hence, it would be very useful to be able to tap into the knowledge of a larger subset of the user population to generate recommendations.

Trust metrics compute an estimate of how much a user should trust another user, based on the existing trust relations between other users in the network. Various types of trust metrics exist in the literature; we refer to [68] for a good overview. In that paper, Ziegler and Lausen classify trust metrics along three dimensions: group versus scalar metrics, centralized versus distributed approaches, and global versus local metrics. The first dimension refers to the way trust relations are evaluated, while the second classification is based on the place where the trust estimations are computed. The last dimension refers to the network perspective: trust metrics can take into account all users and trust relationships between them when computing a trust estimation (see e.g. [28, 44, 55]), or only rely on a part of the trust network, hence taking into account personal bias (e.g. [15, 21, 35]). The trust-enhanced techniques of Section 3 belong to the latter type.

20.2.2.1 Propagation

Trust metrics usually incorporate techniques that are based on the assumption that trust is somehow transitive. We call these techniques trust propagation strategies. Let us illustrate this with Figure 20.1: if user a trusts user b (whom we call a trusted third party, or TTP for short), and TTP b trusts user c, then it is reasonable to assume

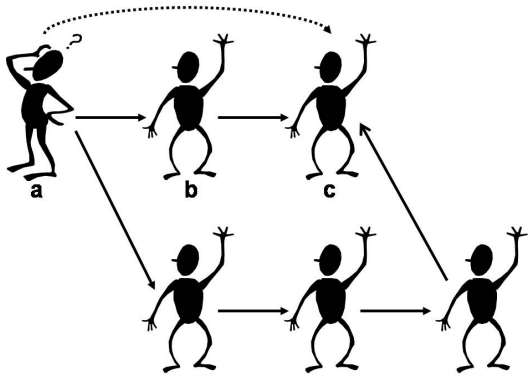

Fig. 20.1: Propagation example

that *a* should trust *c* to a certain degree. This basic propagation strategy is known as atomic direct propagation, and is the type that we will focus on in the remainder of this chapter[5]. However, trust is not always transitive. For instance, if Jane trusts Alice to give her a good-looking haircut and Alice trusts John to fix her bicycle, this does not imply that Jane trusts John to fix bicycles, nor to give a nice haircut.

But, in the same context/scope, and under certain conditions, trust can be transitive [26]. Suppose e.g. that Jane is new in town and wants to have a haircut. Jane trusts that Alice can find a good hairdresser, while Alice trusts Mariah to be a good hairdresser. Hence, Jane can trust Mariah to be a good hairdresser. This example also shows us that a distinction must be made between trust in a user's competence to assess the trustworthiness of a user (functional trust, Alice trusting Mariah), or trust in a user's competence to recommend/evaluate a good recommender agent (referral trust, Jane trusting Alice) [1, 26]. As explained in [26], it is the referral part that allows trust to become transitive. A propagation path can then be seen as a transitive chain of referral trust parts, which ends with one functional trust scope.

When dealing with trust only, in a probabilistic setting, multiplication is very often used as the standard propagation operator, see for instance [55]. This is also the case in gradual settings [3, 15, 21], but there is a wider spectrum of propagation operators available, dependent on the goal or the spirit of the application. This is illustrated by the following example.

[5] For a discussion of other trust propagation strategies, such as cocitation, transpose trust, or coupling, we refer to [21].

Example 20.1. Suppose that, on a scale from 0 to 1, user a trusts user b to the degree 0.5, and that b trusts user c to the degree 0.7. Then, in a probabilistic setting (using standard multiplication), trust propagation yields 0.35. In a fuzzy logic approach however, the final trust estimate depends on the choice of the operator: for instance, the rationale that a propagation chain is only as strong as its weakest link leads to the use of the minimum as propagation operator, hence yielding 0.5 as the propagated trust estimate. The use of the Łukasiewicz conjunction operator on the other hand, i.e. $\max(t_1 + t_2 - 1, 0)$, will yield 0.2. Like with multiplication, this propagated trust value reflects the individual influences of both composing links, as opposed to only the weakest link.

Other trust propagation work includes techniques based on fuzzy if-then rules [31, 59], on the theory of spreading activation models (Ziegler and Lausen's Appleseed [68]), or on the semantic distance between a TTP's trust and a user's perception of the TTP's trust [1].

Of course, not all propagation paths have the same length. In Figure 20.1 e.g., there are two paths leading from the source user a to the target user c. If we suppose that all trust links in the network denote complete trust, then intuitively we feel that the estimated trust of the second propagation path should be lower than that of the first path, since we are heading further away from the source user. This idea of 'trust decay' [20] is often implemented in propagation strategies. For instance, in Ziegler's approach this is incorporated through a spreading factor [68], Golbeck only takes into account shortest paths and ignores all others [15], and in applications that only work with binary trust (instead of gradual), Massa determines the propagated trust based on a user's distance from a fixed propagation horizon [35].

In the case of atomic direct propagation, if a trusts b and b trusts c, a might trust c to a certain degree. Analogously, if a trusts b and b distrusts c, it seems clear that a should somehow distrust c. However, the picture gets more complicated when we also allow distrust as the first link in a propagation chain. For example, if a distrusts b and b distrusts c, there are several options for the trust estimation of a in c: a possible reaction is to infer that a should trust c, since a might think that distrusted acquaintances of users he distrusts are best to be trusted ('the enemy of your enemy is your friend'). Or a should distrust c because a thinks that someone that is distrusted by a user that he distrusts certainly must be distrusted. Yet another interpretation of distrust propagation is to ignore information coming from a distrusted user b, because a might decide not to take into account anything that a distrusted user says.

Guha et al. call the second strategy additive distrust propagation, and the first multiplicative distrust propagation [21]. They discuss the negative side effects of multiplicative propagation (also see [68]), but conclude that it cannot be ignored because it has some philosophical defensibility. Besides Guha et al., other researchers also proposed operators that adhere to the first strategy, such as Victor et al.'s approach using fuzzy logic concepts [62] or Jøsang et al.'s opposite belief favouring discount operator [27]. Examples of the last strategy can be found in [21, 27, 62].

Example 20.2. Like with trust propagation, approaches to distrust propagation are intimately linked to the representations of trust and distrust at hand. Let us assume the use of a couple (t,d) with a trust degree t and a distrust degree d, both in $[0,1]$. In this representation, $(1,0)$ corresponds to full trust, $(0,1)$ corresponds to full distrust, and $(0,0)$ corresponds to full ignorance, or full lack of knowledge. Gradual values such as in $(0.5,0.2)$ denote partial trust 0.5, partial distrust 0.2 and partial lack of knowledge $1-0.5-0.2=0.3$. Assume that the trust score of user a in user b is (t_1,d_1) and, likewise, that the trust score of user b in user c is (t_2,d_2). The trust score (t_3,d_3) of user a in user c can then be calculated as follows [62]:

$$(t_3,d_3) = (t_1 \times t_2, t_1 \times d_2)$$

This propagation strategy reflects the attitude of listening to whom you trust and not deriving any knowledge through a distrusted or unknown third party. Below are some examples of propagated trust scores. Each row correspond to a possible trust score of a in b, each column to a trust score of b in c, and the corresponding table entry contains the propagated trust score of a in c.

	(0.0,0.0)	(0.0,1.0)	(1.0,0.0)	(0.5,0.2)
(0.0,0.0)	(0.0, 0.0)	(0.0,0.0)	(0.0,0.0)	(0.0,0.0)
(0.0,1.0)	(0.0, 0.0)	(0.0,0.0)	(0.0,0.0)	(0.0,0.0)
(1.0,0.0)	(0.0,0.0)	(0.0,1.0)	(1.0,0.0)	(0.5,0.2)
(0.5,0.2)	(0.0,0.0)	(0.0, 0.5)	(0.5,0.0)	(0.25,0.1)

In [25] the same propagation technique is used to combine pairs of beliefs and disbeliefs. Furthermore, subtracting the distrust degree from the trust degree, the propagated trust score collapses to $t_1 \times (t_2 - d_2)$, a propagation scheme proposed in [21].

Example 20.3. Alternatively, the trust score (t_3,d_3) of user a in user c can be calculated as [62]:

$$(t_3,d_3) = (t_1 \times t_2 + d_1 \times d_2 - t_1 \times t_2 \times d_1 \times d_2, t_1 \times d_2 + d_1 \times t_2 - t_1 \times d_2 \times d_1 \times t_2)$$

In this propagation strategy, t_3 is computed as the probabilistic sum of $t_1 \times t_2$ and $d_1 \times d_2$, while d_3 is the probabilistic sum of $t_1 \times d_2$ and $d_1 \times t_2$. The underlying assumption is that a distrusted user is giving the wrong information on purpose. Hence user a trusts user c if a trusted third party tells him to trust c, *or*, if a distrusted third party tells him to distrust c (i.e. the enemy of your enemy is your friend). Subtracting the distrust degree from the trust degree yields $(t_1 - d_1) \times (t_2 - d_2)$, a distrust propagation scheme put forward in [21]. Below are some examples of propagated trust scores.

	(0.0,0.0)	(0.0,1.0)	(1.0,0.0)	(0.5,0.2)
(0.0,0.0)	(0.0, 0.0)	(0.0,0.0)	(0.0,0.0)	(0.0,0.0)
(0.0,1.0)	(0.0, 0.0)	(1.0,0.0)	(0.0,0.1)	(0.2,0.5)
(1.0,0.0)	(0.0,0.0)	(0.0,1.0)	(1.0,0.0)	(0.5,0.2)
(0.5,0.2)	(0.0,0.0)	(0.2, 0.5)	(0.5,0.2)	(0.28,0.2)

All these approaches illustrate the fact that, so far, no consensus has yet been reached on how to propagate distrust. Different operators yield different results depending on the interpretation, thus revealing part of the complex problem of choosing an appropriate propagation scheme for the application at hand.

20.2.2.2 Aggregation

Besides propagation, a trust metric must also include an aggregation strategy. After all, in large networks it will often be the case that not one, but several paths lead to the user for whom we want to obtain a trust estimate. When this is the case, the trust estimates that are generated through the different propagation paths must be combined into one aggregated estimation; see for instance the situation depicted in Figure 20.2.

Metrics that only work with trust mostly use classical aggregation operators such as the minimum, maximum, weighted sum, average, or weighted average [1, 3, 19, 28, 43, 44, 55]. The main benefit of weighted operators is that they give us the opportunity to consider some sources (TTPs or propagation paths) as more important than others. In other words, weighted operators provide a way to model the aggregation process more flexibly.

Aggregation of both trust and distrust has not received much attention so far. Only Jøsang et al. have proposed three aggregation operators (called consensus operators) for the subjective logic framework [27]; however, they assume equally important users.

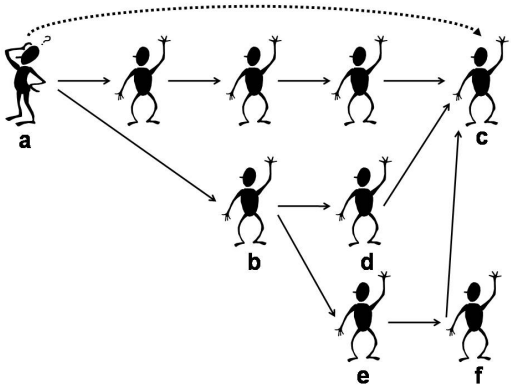

Fig. 20.2: Aggregation example

Note that propagation and aggregation very often must be combined together, and that the final trust estimation might depend on the way this is implemented. Let us take a look at Figure 20.2. There are two ways for user a to obtain a trust estimate about user c from user b. The first possibility is to propagate trust to agent c, i.e., to apply a propagation operator on the trust from b to d and from d to c, and to apply one from b to e, from e to f, and from f to c, and then to aggregate the two propagated trust results. In this scenario, trust is first propagated, and afterwards aggregated (i.e., first propagate then aggregate, or FPTA). A second possibility is to follow the opposite process, i.e., first aggregate and then propagate (FATP). In this scenario, the TTP b must aggregate the estimates that he receives via d and e, and pass on the new estimate to a. It is easy to see that in the latter case the agents/users in the network receive much more responsibility than in the former scenario, and that the trust computation can be done in a distributed manner, without agents having to expose their personal trust and/or distrust information.

Example 20.4. In Figure 20.2 there are three different paths from a to c. Assume that all trust weights on the upper chain are 1, except for the last link which has a trust weight of 0.9. Hence, using multiplication as propagation operator, the propagated trust value resulting from that chain is 0.9. Now, suppose that a trusts b to degree 1, and that b trusts d to the degree 0.5 and e to the degree 0.8. That means that the propagated trust value over the two chains from a to c through b are $1 \times 0.5 \times 0.4 = 0.2$ and $1 \times 0.8 \times 0.6 \times 0.7 \approx 0.34$ respectively. Using the classical average as aggregation operator, FPTA yields a final trust estimate of $(0.9 + 0.2 + 0.34)/3 = 0.48$. On the other hand, if we would allow b to first aggregate the information coming from his trust network, then b would pass the value $(0.2+0.34)/2 = 0.27$ on to a. In a FATP strategy, this would then be combined with the information derived through the upper chain in Figure 20.2, leading to an overall final trust estimate of $(0.9+0.27)/2 \approx 0.59$.

20.3 Trust-Enhanced Recommender Systems

The second pillar of trust-enhanced recommendation research is the recommender system technology. Recommender systems are often used to accurately estimate the degree to which a particular user (from now on termed the target user) will like a particular item (the target item). These algorithms come in many flavours [2, 54]. Most widely used methods for making recommendations are either content-based (see Chapter 3) or collaborative filtering methods (see Chapter 5). Content-based methods suggest items similar to the ones that the user previously indicated a liking for [56]. Hence, these methods tend to have their scope of recommendations limited to the immediate neighbourhood of the user's past purchase history or rating record for items. For instance, if a customer of a DVD rental sevice so far has only ordered romantic movies, the system will only be able to recommend related items, and not explore other interests of the user. Recommender systems can be improved

significantly by (additionally) using collaborative filtering, which typically works by identifying users whose tastes are similar to those of the target user (i.e., neighbours) and by computing predictions that are based on the ratings of these neighbours [53].

In the following section, we discuss the weaknesses of such classical recommender systems and illustrate how they can be alleviated by incorporating a trust network among the users of the system. These advanced, trust-based recommendation techniques adhere closest to the collaborative filtering paradigm, in the sense that a recommendation for a target item is based on ratings by other users for that item, rather than on an analysis of the content of the item. A good overview of classic and novel contributions in the field of trust systems, and trust-aware recommender systems in particular, can be found in the book edited by Golbeck [17].

20.3.1 Motivation

Despite significant improvements on recommendation approaches, some important problems still remain. In [37], Massa and Avesani discuss some of the weaknesses of collaborative filtering systems. For instance, users typically rate or experience only a small fraction of the available items, which makes the rating matrix very sparse (since a recommender system often deals with millions of items). For instance, a particular data set from Epinions contains over 1 500 000 reviews that received about 25 000 000 ratings by more than 160 000 different users [61]. Due to this data sparsity, a collaborative filtering algorithm experiences a lot of difficulties when trying to identify good neighbours in the system. Consequently, the quality of the generated recommendations might suffer from this. Moreover, it is also very challenging to generate good recommendations for users that are new to the system (i.e., cold start users), as they have not rated a significant number of items and hence cannot properly be linked with similar users. Thirdly, because recommender systems 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 [67]. For instance, 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. Finally, Sinha and Swearingen [57, 58] have shown that users prefer more transparent systems, and that people tend to rely more on recommendations from people they trust ('friends') than on online recommender systems which generate recommendations based on anonymous people similar to them.

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) has been found. Trust-enhanced recommender systems try to simulate this behaviour, as depicted in Figure 20.3: once a path to a recommender is found, the system can combine that recommender's judgment with available trust

information (through trust propagation and aggregation) to obtain a personalized recommendation. In this way, a trust network allows to reach more users and more items. In the collaborative filtering setting in Figure 20.4, users a and b will be linked together because they have given similar ratings to certain items (among which i_1), and 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 c) 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 the users of the recommender system.

The solid lines in Figure 20.4 denote trust relations between user a and user b, and between b and user c. While in a scenario without a trust network a collaborative filtering system is not able to generate a prediction about i_3 for user a, this could be solved in the trust-enhanced situation: 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 the outcome would indicate that agent a should highly trust c, then i_3 might become a good recommendation for a, and will be highly ranked among the other recommended items. This simple example illustrates that augmenting a recommender system by including trust relations can help solving the sparsity problem. Moreover, a trust-enhanced system also alleviates the cold start problem: it has been shown that by issuing a few trust statements, compared to a same amount of rating information, the system can generate more, and more accurate, recommendations [35]. 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 explicitly indicated that he trusts the adversary to a certain degree. Finally, the functioning of a trust-enhanced system (e.g. the concept of trust propagation) is intuitively more understandable for the users than the classical 'black box' approaches. A nice example is Golbeck's FilmTrust system [16] which asks its users to evaluate their acquaintances based on their movie taste, and accordingly uses that information to generate personalized predictions.

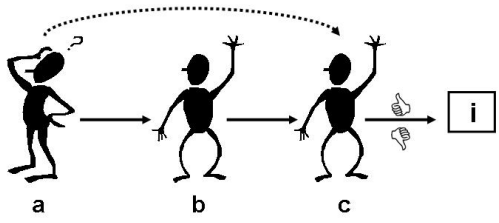

Fig. 20.3: Recommending an item

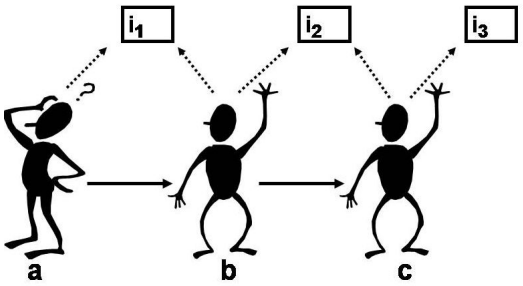

Fig. 20.4: Trust relations in recommender systems

20.3.2 State of the Art

All these examples illustrate that establishing a trust network among the users of a recommender system may contribute to its success. Hence, unsurprisingly, some attempts in this direction have already been made, see for example [15, 23, 30, 37, 46, 49, 51]. Trust-enhanced recommender systems can roughly be divided into two classes, according to the way the trust values are obtained. The first group uses information coming from a trust network that is generated by the direct input of the users, i.e., by explicitly issuing trust statements. Examples can be found in [16, 23, 37]. Such a strategy allows to use trust propagation and aggregation in the network to infer the final trust values that are needed in the recommender algorithm. On the other hand, the second group does not require the user to estimate the trust in his acquaintances. Instead, trust values are computed automatically, for instance based on a user's history of making reliable recommendations [30, 46], or based on transitivity rules for user-to-user similarity [49].

In the behavioral literature, the concept of trust is well defined; see for example Mayer et al.'s framework in which ability, benevolence, integrity and propensity to trust are determined as its key factors [40], or McAllister's work that distinguishes between cognition-based and affect-based trust [41]. However, in the recommendation research area, trust is often used as an umbrella term for a wide range of relationships between people, especially when dealing with automatic computation of trust values. In these cases, trust is being used to denote a variety of concepts, ranging from perceived similarity of tastes, over reputation, to the assessment of a user's competence.

In Section 20.4 we further discuss this in more detail ; in this section, we focus on the basics of both strategies (i.e., mining a trust network and automatic computation of trust values), and illustrate the techniques with representative work in each class.

20.3.2.1 Mining a Trust Network

The most common trust-enhanced recommender strategies ask their users to explicitly issue trust statements about other users. Take for instance Moleskiing [7], a ski mountaineering community site which uses FOAF-files that contain trust information on a scale from 1 to 9 [19], or the e-commerce site Epinions.com which orders reviews based on a trust network that it maintains by asking its users to indicate which members they trust (i.e., their personal web of trust) or distrust (block list). Another well-known example is Golbeck's FilmTrust [16], an online social network combined with a movie rating and review system in which users are asked to evaluate their acquaintances' movie tastes on a scale from 1 to 10.

All these systems exploit the relations in the trust network to determine which opinions or ratings should weigh more or less in the recommendation process. In other words, this group of algorithms uses the trust estimates (obtained by propagation and aggregation) as weights in the decision process. This weighting can be done in several ways. In this section, we focus on the two most commonly used strategies, namely classical weighted average and adaptations of the collaborative filtering mechanism, and illustrate each of them with one well-known state-of-the-art implementation.

Trust-based weighted mean In a recommender system without a trust network, a simple recommendation algorithm that needs to estimate how well a target user will like a target item i can compute the average rating for i by taking into account the ratings $r_{u,i}$ from all the system's users u who are already familiar with i. This baseline recommendation strategy can be refined by computing a *trust-based weighted mean*. In particular, by including trust values $t_{a,u}$ that reflect the degree to which the raters u are trusted, the algorithm allows to differentiate between the sources. In fact, it is only natural to assign more weight to ratings of highly trusted users. The formula is given by Equation (20.1), in which $p_{a,i}$ denotes the predicted rating of target item i for target user a, and R^T represents the set of users who evaluated i and for which the trust value $t_{a,u}$ exceeds a given threshold.

$$p_{a,i} = \frac{\sum_{u \in R^T} t_{a,u} r_{u,i}}{\sum_{u \in R^T} t_{a,u}} \tag{20.1}$$

TidalTrust This formula is at the heart of Golbeck et al.'s recommendation algorithm [15]. The novelty of this algorithm mainly lies in the way the trust estimates $t_{a,u}$ are inferred; a trust metric that they have called *TidalTrust*. In [18], the authors give an overview of the observations that have lead to the development of TidalTrust. In each experiment, they ignored an existing trust relation from a user a to a user c, and focused on all paths that connect a to c. In short, by comparing the propagated trust results from these paths with the original, hidden, trust value, they

noticed that (1) shorter propagation paths yield more accurate trust estimates, and that (2) paths containing higher trust values yield better results too.

Hence, taking into account the first observation, only allowing shorter paths should yield the best results. However, in some cases only a few users will be reachable if a limit is set on the path length. This trade-off is incorporated through a variable path length limit: the shortest path length that is needed to connect the target user with a user u that has rated the item (i.e., a rater) becomes the path depth of the algorithm. Like this, the depth of the breadth-first search varies from one computation to another.

One way of addressing the second observation (higher trust values on the path yield better trust estimates) is to limit the information such that it only comes from the most trusted users. However, every user has its own behaviour for issuing trust values (one user may give the maximum value quite often while another one never does), and in addition, it will often be the case that only a few paths contain the same high trust value. This is why Golbeck et al. opted to incorporate a value that represents the path strength (i.e., the minimum trust rating on a path), and to compute the maximum path strength over all paths leading to the raters. This maximum (*max*) is then chosen as the minimum trust threshold for participation in the process.

The TidalTrust formula is given by Equation (20.2), in which $WOT^+(a)$ represents the set of users for whom a's trust statement exceeds the given threshold *max*. This means that each user in the process computes its trust in another user as a weighted mean, and only takes into account information from users that he has rated at least as high as *max*.

$$t_{a,u} = \frac{\sum_{v \in WOT^+(a)} t_{a,v} t_{v,u}}{\sum_{v \in WOT^+(a)} t_{a,v}} \tag{20.2}$$

TidalTrust is a recursive algorithm; the trust value $t_{a,u}$ is recursively computed as the weighted mean of trust values $t_{v,u}$ for all TTPs v that are the first link on the shortest path from a to u. The users assure that the maximum path depth is not exceeded by keeping track of the current path length. Note that this algorithm belongs to the class of gradual trust approaches and is an example of a local trust metric.

Golbeck et al. have shown that using trust-based weighted mean in combination with TidalTrust does not necessarily offer a general benefit over computing the average or applying collaborative filtering, but that it does yield significantly more accurate recommendations for users who disagree with the average rating for a specific item (see e.g. [15, 18]).

Trust-based collaborative filtering Whereas Golbeck's approach is an example of a weighted average implementation, another class of trust-enhanced systems is tied more closely to the *collaborative filtering* algorithm. In collaborative filtering, a rating of target item i for target user a can be predicted using a combination of the

ratings of the neighbours of a (similar users) that are already familiar with item i [53]. The classical formula is given by Equation (20.3).

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u \in R^+} w_{a,u}(r_{u,i} - \bar{r}_u)}{\sum_{u \in R^+} w_{a,u}} \tag{20.3}$$

The unknown rating $p_{a,i}$ for item i and target user a is predicted based on the mean $\bar{r}_a$ of ratings by a for other items, as well as on the ratings $r_{u,i}$ by other users u for i. The formula also takes into account the similarity $w_{a,u}$ between users a and u, usually calculated as Pearson's Correlation Coefficient (PCC) [22]. In practice, most often only users with a positive correlation $w_{a,u}$ who have rated i are considered. We denote this set by R^+. However, instead of a PCC-based computation of the weights, once can also infer the weights through the relations of the target user in the trust network (again through propagation and aggregation); see Formula (20.4) which adapts Formula (20.3) by replacing the PCC weights $w_{a,u}$ by the trust values $t_{a,u}$. This strategy is also supported by the fact that trust and similarity are correlated, as shown in [69].

$$p_{a,i} = \overline{r_a} + \frac{\sum_{u \in R^T} t_{a,u}(r_{u,i} - \overline{r_u})}{\sum_{u \in R^T} t_{a,u}} \tag{20.4}$$

We call this alternative *trust-based collaborative filtering*. Note that, because the weights are not equal to the PCC, this procedure can produce out of bounds results. When this is the case, $p_{a,i}$ is rounded to the nearest possible rating.

MoleTrust Formula (20.4) is at the basis of Massa et al.'s recommendation algorithm which incorporates a new trust metric, called *MoleTrust* [38]. This metric was mainly devised for testing purposes, which reveals itself in tunable parameters and cycle removing. MoleTrust consists of two phases. In the first stage, cycles in the trust network are removed, while the second stage includes the actual trust computation. Since it is often the case that a large number of trust propagations must be executed in trust experiments (think e.g. of the large test sets from Epinions.com), it is much more efficient to remove trust cycles beforehand, so that every user only needs to be visited once for obtaining a trust prediction.

The removing of the cycles transforms the original trust network into a directed acyclic graph, and hence the trust prediction for $t_{a,u}$ can be obtained by performing a simple graph walk: first the trust of the users at distance 1 is computed (i.e., direct trust information), then the trust of the users at distance 2, etc. Note that because of the acyclic nature of the graph, the trust value of a user at distance x only depends on the already computed trust values of the users at distance $x-1$.

The trust of the users at distance 2 or more is calculated in a way similar to Golbeck et al.'s algorithm, i.e. formula (20.2). However, the details of the breadth-first implementation differ significantly. In TidalTrust, a user u is added to $WOT^+(a)$

Table 20.2: Characteristic features of two state-of-the-art recommendation approaches that mine a trust network to predict a rating for target user a and target item i, based on ratings of other users u for i

	TidalTrust	MoleTrust
propagation	multiplication	multiplication
aggregation	trust-based weighted mean (20.2)	trust-based weighted mean (20.2)
maximum length of propagation path	dynamic (shortest path)	static (horizon)
trust threshold	dynamic (strongest chain)	static
entry requirement for TTP v in propagation process	v is on a shortest path from a to u	v is on a path from a to u within the horizon
prediction of rating	trust-based weighted mean (20.1)	trust-based collaborative filtering (20.4)

only if he is on a shortest path from target user a to target item i. On the other hand, in MoleTrust, $WOT^+(a)$ includes all users who have rated the target item and that can be reached through a direct or propagated trust relation. But trust is not computed for all eternity: before the computation begins, one must assign a value d to the 'propagation horizon' parameter. Like this, only users who are reachable within distance d are taken into account. Another important input parameter of MoleTrust is the trust threshold for participation in the process (unlike the dynamic *max* value in TidalTrust), which is for example set to 0.6 (on a scale from 0 to 1) in the experiments reported in [38].

Note that, analogous to TidalTrust, MoleTrust belongs to the class of gradual local trust metrics. In their experiments, Massa and Avesani have illustrated that MoleTrust provides better trust estimates than global trust metrics such as eBay's[6], especially when it comes down to estimating the trust in controversial users (who are trusted by one group and distrusted by another) [38]. They also showed that MoleTrust yields more accurate predictions for cold start users, compared to a classical collaborative filtering system [35, 36].

Golbeck's and Massa's approach are two typical examples of trust-enhanced recommender techniques that use explicit trust information. Table 20.2 summarizes their most prominent characteristics. Other recommendation approaches that also mine a trust network can be found in, among others, [23, 63].

[6] www.ebay.com

20.3.2.2 Automatic Trust Generation

The algorithms discussed in the previous section require explicit trust input from the users. As a consequence, the applications that use such an algorithm must provide a means to obtain the necessary information; think e.g. of FilmTrust or Moleskiing. However, this might not always be possible or feasible. In such cases, methods that automatically infer trust estimates, without needing explicit trust information, might be a better solution. An example of such a system can be found in [47].

Most commonly, these approaches base their trust generation mechanism on the past rating behaviour of the users in the system. More specifically, deciding to what degree a particular user should participate in the recommendation process is influenced by his history of delivering accurate recommendations. Let us exemplify this with the well-known approach of O'Donovan et al. [46].

Profile- and item-level trust Our intuition tells us that a user who has made a lot of good recommendations in the past can be viewed as more trustworthy than other users who performed less well. To be able to select the most trustworthy users in the system, O'Donovan introduced two trust metrics, viz. *profile-level* and *item-level trust*, reflecting the general trustworthiness of a particular user u, and the trustworthiness of a user u with respect to a particular item i, respectively. Both trust metrics need to compute the correctness of u's recommendations for the target user a. In particular, a prediction $p_{a,i}$ that is generated only by information coming from u (hence u is the sole recommender) is considered correct if $p_{a,i}$ is within ε of a's actual rating $r_{a,i}$.

The profile-level trust $t^P_{a,u}$ for u seen from a's perspective is then defined as the percentage of correct recommendations for a made by u. Remark that this is a very general trust measure; in practice it will often occur that u perfoms better in recommending a set of specific items. To this aim, O'Donovan also proposed the more fine-grained item-level trust $t^i_{a,u}$, which measures the percentage of recommendations for item i that were correct. Hence, in such automated approaches, trust values are not generated via trust propagation and aggregation, but are based on the ratings that were given in the past. Remark that O'Donovan's methods, too, are local trust metrics. The way the values are obtained can be seen as probabilistic.

Trust-based filtering Similar to other trust-enhanced techniques, the values that are obtained through the trust metric are used as weights in the recommendation process. Just like Massa, O'Donovan et al. focus on trust-based adaptations of collaborative filtering. In [46] they investigate several options, such as combining the obtained trust values with PCC information. An alternative to this scheme is to use trust values as a filter, so that only the most trustworthy neighbours participate in the recommendation process. This strategy is called *trust-based filtering*, see Formula (20.5) in which $w_{a,u}$ denotes the PCC and $R^{T+} = R^T \cap R^+$.

$$p_{a,i} = \overline{r_a} + \frac{\sum_{u \in R^{T+}} w_{a,u}(r_{u,i} - \overline{r_u})}{\sum_{u \in R^{T+}} w_{a,u}} \tag{20.5}$$

In other words, only users whose item/profile-level trust exceeds a certain threshold, and that have a positive correlation with a, are taken into account.

In [46], O'Donovan and Smyth showed that trust-based filtering achieves better accuracy than collaborative filtering in terms of average errors. Moreover, the algorithm based on profile-level trust yields lower errors than collaborative filtering in nearly 70% of all prediction cases.

O'Donovan's method is a representative example in the group of strategies that use automatic trust generation. A related approach can be found in [30], which works with an utilitarian measure instead of a binary correctness function.

20.3.3 Empirical Comparison

One question that stands out is which of the state-of-the-art approaches discussed above performs best in practice. Basically, so far, researchers in the trust-based recommender field introduced their own new algorithms and evaluated these on their own applications and/or data sets, without including a comparison of other trust-enhanced approaches based on the same data set/application. Therefore, in the remainder of this section, we provide a head-to-head comparison of the performance that the previously discussed trust-enhanced techniques can achieve on one and the same data set. We focus on Golbeck's trust-based weighted mean with TidalTrust (Eq. (20.1)), Massa's trust-based collaborative filtering with MoleTrust (Eq. (20.4)), and O'Donovan's trust-based filtering (Eq. (20.5)). Since our goal is to compare all techniques on the same data sets and to investigate the influence of trust propagation, we have chosen not to implement O'Donovan's automatic trust generation strategy, but to mine the same trust network as the other two strategies. Although O'Donovan et al. do not use trust propagation in their experiments [46], it is of course possible to do so. Since there is no explicit use of trust values in (20.5), we only need to specify how propagation enlarges R^{T+} (see below).

20.3.3.1 Data Sets

The data sets we use in our experiments are obtained from Epinions.com, a popular e-commerce site where users can write reviews about consumer products and assign a rating to the products and the reviews. Two Epinions data sets are often used for experimenting with trust-enhanced recommender systems. The first one was collected by Massa and Bhattacharjee [39] in a 5-week crawl and contains 139 738 products that are rated by 49 290 users in total; the consumer products are rated on a scale

from 1 to 5. The second data set was compiled by Guha et al. [21]: this large data set contains 1 560 144 reviews that received 25 170 637 ratings by 163 634 different users. The reviews are evaluated by assigning a helpfulness rating which ranges from 'not helpful' (1/5) to 'most helpful' (5/5). This data set does not contain any information about consumer products and product ratings, but works with reviews and review ratings instead; in other words, for this data set, we discuss and evaluate a 'review recommender system'. Hence, in the context of Guha's set, an item denotes a review of consumer goods, whereas for the crawled data set an item denotes a consumer product.

In our experiments we focus on the number of recommendations/predictions that can be generated by the systems and on the prediction errors, for random items as well as controversial items. The latter are the most challenging items for a recommender system, since it is much harder to predict a score for an item that has received a variety of high and low scores, reflecting disagreement about the item. More than in any other case, a recommendation for a user needs to be truly personalized when the target item under consideration is controversial; i.e., when an item has both 'ardent supporters' and 'motivated adversaries', with no clear majority in either group. In [63], Victor et al. explain why classical standard deviation is not sufficient to detect the true controversial items in a data set, and propose a new measure to define the controversiality level of a particular item. Their methodology leads to 1 416 controversial items in Guha's data set, and 266 in Massa's data set. We refer to [63] for more details about the controversiality computation. To compare the performance achieved for controversial items (CIs) with the performance that can be obtained in general, we also present the average coverage and accuracy for 1 416 and 266 randomly selected 'popular' items (RIs) (that have been evaluated at least 20 times, analogous to the controversial items).

Epinions allows users to evaluate other users based on the quality of their reviews, and to provide trust and distrust evaluations in addition to ratings. The fact that both data sets contain explicit trust information from the users makes them very appropriate to study issues in trust-enhanced recommender systems. Users can evaluate other users by including them in their WOT (i.e. a list of reviewers whose reviews and ratings were consistently found to be valuable[7]), or by putting them in their block list (a list of authors whose reviews were consistently found to be offensive, inaccurate or low quality[7], thus indicating distrust). In Guha's data set, the trust evaluations make up an Epinions WOT graph consisting of 114 222 users and 717 129 non self-referring trust relations. Massa's data set contains information on 49 288 users who issued or received 487 003 trust statements in total.

Note that the data sets only contain binary trust values, hence in our experiments $t_{a,u}$ in (20.1), (20.4) and (20.5) can take on the values 0 (absence of trust) and 1 (full presence) only. This limitation leads to alterations of some of the trust-basesd algorithms; e.g., Formula (20.1) reduces to the classical average. For simplicity, we

[7] See www.epinions.com/help/faq/

only consider one-step propagation in this paper. This means that for the propagated versions of (20.4) and (20.5), we consider chains of length 1 and 2, whereas for (20.2) we only consider chains of length 2 when there are no shorter chains available. These two simplifications put a restriction on our empirical comparison, because we cannot analyse the algorithms exactly as they were meant/designed to be.

20.3.3.2 Coverage

Coverage refers to the number of target user - target item pairs for which a prediction can be generated. A classical way to measure the coverage of a recommender system is by using the leave-one-out method, which consists of hiding a rating and trying to predict its hidden value. The coverage of a specific algorithm then refers to the amount of computable predictions $p_{a,i}$ versus the number of leave-one-out experiments to perform (i.e., the number of ratings available in the data set). For Formula (20.3) we call $p_{a,i}$ computable if there is at least one user u for which the PCC $w_{a,u}$ can be calculated, while for Formulas (20.1) and (20.4) a computable $p_{a,i}$ means that there is at least one user u for which the (propagated) trust estimate $t_{a,u}$ can be calculated. Finally, for Formula (20.5), predictions are possible when at least one user u is found for which the PCC can be computed and $t_{a,u}$ is 1.

Table 20.3 shows the coverage (% COV) for controversial items (CIs) and randomly selected items (RIs) in Guha's and Massa's data sets. The first four rows cover baseline strategies (B1)–(B4). The first baseline strategy is a system that always predicts 5/5 (B1), since this is the predominant score for items in Epinions. The second system computes the average received rating for the target item (B2), while the third one yields the average rating given by target user a (B3). The latter method will score well in a system where the users have a rating behaviour with little variation. Finally, the last baseline returns a random helpfulness score between 1 and 5 (B4).

In general, baselines (B1), (B2) and (B4) achieve maximal coverage for both controversial and randomly selected items: (B1) and (B4) do not rely on any additional (trust or PCC) information, and since the items in our experiments are evaluated at least 20 times, it is always possible to compute (B2). With (B3), in those cases in which the target user rated only one item, his average rating is lacking, so a prediction cannot be generated.

For the other algorithms in Table 20.3, the numbers in the first column refer to the corresponding recommendation formulas given above. For the trust-enhanced approaches, we distinguish between experiments that did not use propagated trust information (higher rows) and those that did (bottom rows). We only consider one-step propagation: for (P1) and (P4), we maintained the propagation strategy used in

TidalTrust and MoleTrust[8] respectively, while for (P5) we added a user to R^T if he belongs to the WOT of the target user a, or is directly trusted by a WOT member of a.

Without propagation, it is clear that the coverage of the collaborative filtering algorithm is superior to that of the others, and approaches the maximal value. This is due to the fact that PCC information is, in general, more readily available than direct trust information: there are normally more users for which a positive correlation with the target user a can be computed than users in a's WOT. On the other hand, trust-based filtering (20.5), which also uses PCC weights, is the most demanding strategy because it requires users in a's WOT who have already rated two other items in common with a (otherwise the PCC can not be computed). In between these extremes, the coverage for TidalTrust (20.1) is a bit higher than that of MoleTrust (20.4) because the latter can only generate predictions for target users who have rated at least two items, otherwise the average rating for the target user can not be computed).

This ranking of approaches in terms of coverage still applies when propagated trust information is taken into account, but note that the difference with collaborative filtering has shrunk considerably. In particular, thanks to trust propagation, the coverage increases with about 25% (10%) for controversial (randomly selected) items in the first set, and more than 30% in the second set.

For Guha's data set, the coverage results for controversial items are significantly lower than those for randomly selected items. This is due to the fact that, on average, controversial items in this data set receive less ratings than randomly selected items, which yields less leave-one-out experiments per item, but also a smaller chance that such an item was rated by a user with whom the target user a has a positive PCC, or by a user that a trusts. This also explains the lower coverage results for the nontrivial recommendation strategies. The same observations cannot be made for Massa's data set: on average, the CIs receive more ratings than the RIs (21 131 vs. 12 741). This explains the somewhat lower coverage performance of the algorithms on the random item set.

Also remark that the coverage results for Massa's data set are significantly lower in general than those for Guha's; (20.1), (20.4) and (20.5) achieve a coverage that is at least 20% worse. Users in Guha's data set rate much more items than users in Massa's data set, which yields less users who have rated the same items, i.e., neighbours (through trust or PCC) that are needed in the computation.

20.3.3.3 Accuracy

As with coverage, the accuracy of a recommender system is typically assessed by using the leave-one-out method, more in particular by determining the deviation between the hiding ratings and the predicted ratings. In particular, we use two well-known measures, viz. mean absolute error (MAE) and root mean squared error

[8] Note that we incorporate Massa et al.'s horizon-based strategy for binary trust settings [35].

Table 20.3: Performance trust-based recommender algorithms

		Guha et al.'s data set						*Massa et al.'s data set*					
		Controversial items (CIs)			*Randomly selected items (RIs)*			*Controversial items (CIs)*			*Randomly selected items (RIs)*		
	ALGORITHM	*% COV*	*MAE*	*RMSE*	*% COV*	*MAE*	*RMSE*	*% COV*	*MAE*	*RMSE*	*% COV*	*MAE*	*RMSE*
(B1)	Base: score 5	100	1.45	1.96	100	0.16	0.51	100	1.94	2.46	100	1.05	1.62
(B2)	Base: average score for item	100	1.25	1.34	100	0.18	0.40	100	1.35	1.51	100	0.82	1.06
(B3)	Base: average score of user	99	1.23	1.58	100	0.36	0.50	98	1.43	1.78	99	0.95	1.22
(B4)	Base: random score	100	1.61	2.02	100	1.92	2.37	100	1.66	2.08	100	1.68	2.10
(20.3)	Collaborative filtering	94	0.96	1.13	98	0.19	0.38	81	1.34	1.58	79	0.84	1.12
(20.1)	Trust-based weighted mean	63	0.86	1.20	89	0.13	0.35	41	1.33	1.70	34	0.87	1.24
(20.4)	Trust-based collaborative filtering	63	0.87	1.16	89	0.17	0.35	40	1.32	1.65	34	0.86	1.19
(20.5)	Trust-based filtering	60	0.86	1.16	86	0.16	0.36	25	1.35	1.71	22	0.85	1.18
(P1)	Propagated Trust-based weighted mean	88	0.91	1.22	97	0.15	0.38	76	1.37	1.69	72	0.90	1.23
(P4)	Propagated Trust-based collaborative filtering	88	0.99	1.16	97	0.19	0.37	76	1.32	1.56	72	0.84	1.12
(P5)	Propagated Trust-based filtering	84	0.94	1.13	96	0.18	0.36	57	1.36	1.64	53	0.86	1.16

(RMSE) [22]. The first measure considers every error of equal value, while the latter one emphasizes larger errors. Since reviews and products are rated on a scale from 1 to 5, the extreme values that MAE and RMSE can reach are 0 and 4. Even small improvements in RMSE are considered valuable in the context of recommender systems. For example the Netflix prize competition[9] offers a $1 000 000 reward for a reduction of the RMSE by 10%.

The MAE and RMSE reported in Table 20.3 is overall higher for the controversial items than for the randomly selected items. In other words, generating good predictions for controversial items is much harder than for randomly chosen items. This applies to all the algorithms, but most clearly to the baseline strategies (except (B4)). While in Massa's data set all algorithms adjust themselves in more or less the same way, in Guha's data set (B1) and (B2) clearly experience more difficulties when generating predictions for controversial items: whereas for random items they are competitive with collaborative filtering and the trust-enhanced approaches, their MAE and RMSE on the controversial item set increase with more than 1 on the rating scale from 1 to 5.

Also note that it is more difficult to generate good recommendations in Massa's data set than in Guha's, for controversial as well as random items. This is due to the higher inherent controversiality level of the former data set.

When focusing on the MAE of the non-baseline approaches for controversial items, we notice that, without propagation, trust-enhanced approaches all yield better results than collaborative filtering[10] (with one exception for trust-based filtering on Massa's CIs), which is in accordance with the observations made in [15, 36]. This can be attributed to the accuracy/coverage trade-off: a coverage increase is usually at the expense of accuracy, and vice versa. It also becomes clear when taking into account trust propagation: as the coverage of the trust-enhanced algorithms nears that of the collaborative filtering algorithm, so do the MAEs.

However, the RMSEs give us a different picture. On the controversial item sets, the RMSE of the trust-enhanced approaches is generally higher than that of collaborative filtering, which does not always occur on the random sets; recall that a higher RMSE means that more large prediction errors occur. One possible explanation for this is the fact that, for controversial items, the set R^T of trusted acquaintances that have rated the target item is too small (e.g., contains only 1 user), and in particular smaller than R^+. This hypothesis is also supported by the fact that with trust propagation (which enlarges R^T) RMSEs rise at a slower rate than the corresponding MAEs. Moreover, it is often the case that the propagated algorithms achieve lower RMSEs than their unpropagated counterparts, see e.g. the results on controversial items in Massa's data set.

[9] See http://www.netflixprize.com/

[10] Note that all the MAE improvements on Guha's data set are statistically significant ($p < 0.000$).

20.3.3.4 Conclusion

The experiments on both Epinions data sets, each with their own characteristics, endorse the same conclusions. For random items, intelligent strategies such as collaborative filtering and trust-based algorithms barely outperform the baselines. However, the baselines fall short in generating good recommendations for controversial items. Trust-enhanced systems perform better in this respect, although there is certainly still room for improvement; remember the higher RMSEs and the fact that trust-based approaches on Massa's CIs yield no visible improvements over collaborative filtering. These findings call for further research on improving the algorithms and identifying specific cases where trust approaches are effective (think e.g. of Massa et al.'s results for cold start users).

The coverage and accuracy results show no clear winner among the three state-of-the-art trust-enhanced strategies proposed by Golbeck et al., Massa et al., and O'Donovan et al. Trust-based collaborative filtering seems to score best on Massa's data set, while trust-based weighted mean and trust-based filtering achieve the best accuracy on Guha's data set; this trend is also confirmed by the results obtained by propagation.

The two data sets contain rating information and trust information, which makes them popular in trust-enhanced recommender experiments. However, they have one shortcoming: the trust values in Epinions are binary, making it impossible to investigate all aspects of the algorithms we discussed in this chapter, since a lot of the existing trust-based approaches are based on the assumption that trust is a gradual concept. Unfortunately, there are no such data sets publicly available.

20.4 Recent Developments and Open Challenges

In the previous sections we have covered the basics of trust modeling, trust metrics, and trust-enhanced recommender systems. In this section, we want to give the reader a foretaste of new directions in the research area of trust-based recommendation systems. This is certainly not meant to be a complete overview, but rather a selection of recent developments in the field. In particular, we will briefly discuss the following issues: alleviating the trust-based cold start problem, visualization of trust-enhanced recommender systems, theoretical foundations for trust-based research, and involving distrust in the recommendation process.

Massa and Avesani have shown that the user cold start problem in classical recommender systems can be alleviated by including a trust network among its users. They demonstrated that, for new users, it is more beneficial to issue a few trust statements (compared to rating some items) in order to get good recommendations from the system [35]. However, Victor et al. have shown that cold start users in the classical sense (who rated only a few items) are very often cold start users in the trust sense as well [61]. Hence, new users must be encouraged to connect to other

users to expand the trust network as soon as possible, but choosing whom to connect to is often a difficult task. Given the impact this choice has on the delivered recommendations, it is critical to guide newcomers through this early stage connection process. In [61] this problem is tackled by identifying three types of key figures in the recommender system's network, viz. frequent raters, mavens and connectors. The authors show that, for a cold start user, connecting to one of the identified key figures is much more advantageous than including a randomly chosen user, with respect to coverage as well as accuracy of the generated recommendations.

Remark that these connection guidance issues link up with the broader problem of trust bootstrapping, i.e., the problem of how to establish initial trust relations in the network. O'Donovan, too, addresses this problem, but in a very different way: he introduces PeerChooser, a new procedure to visualize a trust-based collaborative filtering recommender system [45]. More specifically, PeerChooser visualizes both information coming from the traditional similarity measure PCC, and information coming from the underlying trust-space generated fom the rating data (remember O'Donovan's profile- and item-level trust [46]). One of the main features of the system is its possiblity to extract trust information on the fly, directly from the user at recommendation time. This is done by moving specific icons (representing users in the system) on an interactive interface. In this way, the user can indicate his mood and preferences, thereby actively providing real-time trust information.

There are also other ways to establish trust relations when the information is not explicitly given by the users. Several sources of social data can be consulted, such as online friend and business networks (think e.g. of Facebook or LinkedIn), e-mail communication, reputation systems, etc. In the recommender system literature, they are often lumped together and collectively referred to as trust, although they map onto different concepts: behavioral theory clearly draws a distinction between homophily or cognitive similarity (similarity between people/tastes/etc.), social capital (reputation, opinion leadership), tie strength (in terms of relationship duration and interaction frequency), and trust (see e.g. [40, 41]). Potentially all these social data sources could be incorporated into a (trust-enhanced) recommender system, but so far not much research has been conducted to find out which ones will be most useful [4], and whether these sources would provide similar results as the classical trust-based recommendation approaches discussed in this chapter. In [5], Arazy et al. embark upon this problem and argue that the design of social recommenders should be grounded in theory, rather than making ad hoc design choices as is often the case in current algorithms.

Another recent research direction of a completely different nature is the investigation of the potential of distrust in trust-based recommender systems. Whereas in the trust modeling domain only a few attempts have been made to incorporate distrust, in the recommender domain this is even less so. This is due to several reasons, the most important ones being that very few data sets containing distrust information are available, and that there is no general consensus yet about how to propagate it

and to use it for recommendation purposes. A first experimental evaluation of the effects of involving distrust in the recommendation process is reported in [64]. In this paper, three distrust strategies are investigated, viz. distrust as an indicator to reverse deviations, distrust as a filter for neighbour selection, and distrust as a debugger of a web of trust. The first two strategies are based on the rationale that trust can be used to select similar users (neighbours) in collaborative filtering systems, while the latter strategy has been suggested by various researchers in the field, see e.g. [20, 68]. The results indicate that the first technique is not the line to take. Distrust as a filter and/or debugger looks more promising, but it is clear that much work remains to be done in this nascent research area before one can come to a more precise conclusion.

20.5 Conclusions

In this chapter we have given an introduction to the research area of trust modeling, and illustrated how trust networks can improve the performance of classical recommender systems. We discussed several state-of-the-art implementations of these so-called trust-enhanced recommender strategies, and provided an experimental evaluation of their performance on two data sets from Epinions.com. This comparison in terms of coverage and accuracy did not yield any clear winner, but did show that each of the algorithms has its own merits.

Recommender applications that maintain a social trust network among their users can benefit from trust propagation strategies that have proven to yield a surplus value, whereas in cases where it is not immediately possible to collect explicit trust statements, methods that are able to automatically compute trust values seem the most ideal solution. Of course, these strategies could not have been devised without the appropriate data sets and/or applications to experiment with.

In fact, one of the main difficulties in the trust-enhanced recommender research domain is the lack of publicly available and suitable test data. Hence, it is our hope that in the near future more such data and applications become within reach of every researcher in need of it, and we strongly believe that this will attract and inspire even more people, thereby stimulating the research in this thriving area of trust-based recommendation.

References

1. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. In: Proc. of the 33rd Hawaii International Conference on System Sciences, pp. 1769-1777 (2000)
2. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering **17**, 734–749 (2005)

3. Almenárez, F., Marín, A., Campo, C., García, C.: PTM: A pervasive trust management model for dynamic open environments. In: Proc. of the First Workshop on Pervasive Security, Privacy and Trust, in conjuntion with Mobiquitous (2004)
4. Arazy, O., Elsane, I., Shapira, B., Kumar, N.: Social relationships in recommender systems. In: Proc. of the 17th Workshop on Information Technologies & Systems (2007)
5. Arazy, O., Kumar, N., Shapira, B.: Improving social recommender systems. IT Professional May/June, 31–37 (2009)
6. Artz, D., Gil, Y.: A survey of trust in computer science and the semantic web. Journal of Web Semantics **5**, 58–71 (2007)
7. Avesani, P., Massa, P., Tiella, R.: Moleskiing.it: a trust-aware recommender system for ski mountaineering. International Journal for Infonomics (2005)
8. Cacioppo, J., Berntson, G.: Relationship between attitudes and evaluative space: a critical review, with emphasis on the separability of positive and negative substrates. Psychological Bulletin **115**, 401–423 (1994)
9. Constantinople, A.: An eriksonian measure of personality development in college students. Development Psychology **1**, 357–372 (1969)
10. Cofta, P.: Distrust. In: Proc. of the International Conference on Electronic Commerce, pp. 250-258 (2006)
11. De Cock, M., Pinheiro da Silva, P.: A many-valued representation and propagation of trust and distrust. In: Bloch, I., Petrosino, A., Tettamanzi, A. (eds.) Lecture Notes in Computer Science 3849, pp. 108-113 (2006)
12. Falcone, R., Pezzulo, G., Castelfranchi, C.: A fuzzy approach to a belief-based trust computation. In: Eder, J., Haav, H-M., Kalja, A., Penjam, J. (eds.) Lecture Notes in Artificial Intelligence 2631, pp. 73-86 (2003)
13. Gans, G., Jarke, M., Kethers, S., Lakemeyer, G.: Modeling the impact of trust and distrust in agent networks. In: Proc. of the Third Workshop on Agent-oriented Information Systems, pp. 45-58 (2001)
14. Ginsberg, M.: Multi-valued logics: A uniform approach to reasoning in artificial intelligence. Computational Intelligence **4**, 265–316 (1988)
15. Golbeck, J.: Computing and applying trust in web-based social networks. PhD thesis (2005)
16. Golbeck, J.: Generating predictive movie ratings from trust in social networks. In: Stølen, K, Winsborough, W.H., Martinelli, F., Massacci, F. (eds.) Lecture Notes in Computer Science 3986, pp. 93-104 (2006)
17. Golbeck, J.: Computing with Social Trust. Springer, London (2009)
18. Golbeck, J., Mannes, A.: Using trust and provenance for content filtering on the semantic web. In: Proc. of the WWW06 Models of Trust for the Web Workshop (2006)
19. Golbeck, J., Parsia, B., Hendler, J.: Trust networks on the semantic web. In: Klusch, M., Omicini, A., Ossowski, S., Laamanen, H. (eds.) Lecture Notes in Artificial Intelligence 2782, pp. 238-249 (2003)
20. Guha, R.: Open rating systems. Technical report, Stanford Knowledge Systems Laboratory (2003)
21. Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: Proc. of the World Wide Web Conference, pp. 403-412 (2004)
22. Herlocker, J., Konstan, J., Terveen, L., Riedl, J.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems **22**, 5–53 (2004)
23. Hess, C., Schiedler, C.: Trust-based recommendations for documents. AI Communications **21**, 145–153 (2008)
24. Jøsang, A.: A logic for uncertain probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **9**, 279-311 (2001).
25. Jøsang, A., Knapskog, S.: A metric for trusted systems. In: Proc. of the National Computer Security Conference, pp. 16-29 (1998)
26. Jøsang, A., Gray, E., Kinateder, M.: Simplification and analysis of transitive trust networks. Web Intelligence and Agent Systems **4**, 139–161 (2006)

27. Jøsang, A., Marsh, S., Pope, S.: Exploring different types of trust propagation. In: Stølen, K, Winsborough, W.H., Martinelli, F., Massacci, F. (eds.) Lecture Notes in Computer Science 3986, pp. 179-192 (2006)
28. Kamvar, S., Schlosser, M., Garcia-Molina, H.: The eigentrust algorithm for reputation management in P2P networks. In: Proc. of the World Wide Web Conference, pp. 640-651 (2003)
29. Klir, G., Yuan, B.: Fuzzy sets and systems: theory and applications. Prentice Hall PTR, New Jersey (1995)
30. Lathia, N., Hailes, S., Capra, L.: Trust-based collaborative filtering. In: Karabulut, Y., Mitchell, J., Herrmann, P., Damsgaard Jensen, C. (eds.) IFIP International Federation for Information Processing **263**, 119–134 (2008)
31. Lesani, M., Bagheri, S.: Applying and inferring fuzzy trust in semantic web social networks. In: Kodé, M.T., Lemire, D. (eds.) Semantic Web and Beyond 2, pp. 23-43 (2006)
32. Levien, R.: Attack-resistant trust metrics. In: Golbeck, J. (ed.) Computing With Social Trust, pp. 121-132 (2009)
33. Lewicki, R., McAllister, D., Bies, R.: Trust and distrust: new relationships and realities. Academy of Management Review **23**, 438–458 (1998)
34. Marsh, S., Briggs, P.: Examining trust, forgiveness and regret as computational concepts. In: Golbeck, J. (ed.) Computing With Social Trust, pp. 9-43 (2009)
35. Massa, P., Avesani, P.: Trust-aware collaborative filtering for recommender systems. In: Proc. of the Federated International Conference On The Move to Meaningful Internet, pp. 492-508 (2004)
36. Massa, P., Avesani, P.: Trust-aware recommender systems. In: Proc. of ACM Recommender Systems, pp. 17-24 (2007)
37. Massa, P., Avesani, P.: Trust metrics in recommender systems. In: Golbeck, J. (ed.) Computing with Social Trust, pp. 259-285 (2009)
38. Massa, P., Avesani, P.: Trust metrics on controversial users: balancing between tyranny of the majority and echo chambers. International Journal on Semantic Web and Information Systems **3**, 39–64 (2007)
39. Massa, P., Bhattacharjee, B.: Using trust in recommender systems: an experimental analysis. In: Jensen, C., Poslad, S., Dimitrakos, T. (eds.) Lecture Notes in Computer Science 2995, pp. 221-235 (2004)
40. Mayer, R., Davis, J., Schoorman, D.: An integrative model of organizational trust. The Academy of Management Review **20**, 709–734 (1995)
41. McAllister, D.: Affect- and cognition-based trust as foundations for interpersonal cooperation in organizations. The Academy of Management Journal **38**, 24–59 (1995)
42. Moskovitch, R., Elovici, Y., Rokach, L., Detection of unknown computer worms based on behavioral classification of the host, Computational Statistics and Data Analysis, 52(9):4544–4566 (2008)
43. Mui, L., Mohtashemi, M., Halberstadt, A.: A computational model of trust and reputation. In: Proc. of the 35th Hawaii International Conference on System Sciences, pp. 2431-2439 (2002)
44. Noh, S.: Calculating trust using aggregation rules in social networks. In: Xiao, B., Yang, L., Ma, J., Muller-Schloer, C., Hua, Y. (eds.) Lecture Notes in Computer Science 4610, pp. 361-371 (2007)
45. O'Donovan, J.: Capturing trust in social web applications. In: Golbeck, J. (ed.) Computing With Social Trust, pp. 213-257 (2009)
46. O'Donovan, J., Smyth, B.: Trust in recommender systems. In: Proc. of the 10th International Conference on Intelligent User Interfaces, pp. 167-174 (2005)
47. O'Donovan, J., Smyth, B.: Mining trust values from recommendation errors. International Journal on Artificial Intelligence Tools **15**, 945–962 (2006)
48. O'Reilly, T.: What is web 2.0. Available at http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html (2005)
49. Papagelis, M., Plexousakis, D., Kutsuras, T.: Alleviating the sparsity problem of collaborative filtering using trust inferences. In: Herrmann, P., Issarny, V., Shiu, S. (eds.) Lecture Notes in Computer Science 3477, pp. 224-239 (2005)

50. Petty, R., Wegener, D., Fabrigar, L.: Attitudes and attitude change. Annual Review of Psychology **48**, 609–647 (1997)
51. Pitsilis, G., Marshall, L.: A trust-enabled P2P recommender system. In: Proc. of the 15th Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, pp. 59-64 (2006)
52. Priester, J., Petty, R.: The gradual threshold model of ambivalence: relating the positive and negative bases of attitudes to subjective ambivalence. Journal of Personality and Social Psychology **71**, 431–449 (1996)
53. Resnick, P., Iacovou, N., Suchak, M., Bergstorm, P., Riedl, J.: Grouplens: An open architecture for collaborative filtering of netnews. In: Proc. of Computer Supported Cooperative Work, pp. 175-186 (1994)
54. Resnick, P., Varian, H.R.: Recommender systems. Communications of the ACM **40**, 56–58 (1997)
55. Richardson, M., Agrawal, R., Domingos, P.: Trust management for the semantic web. In: Proc. of the Second International Semantic Web Conference, pp. 351-368 (2003)
56. Schafer, J., Konstan, J., Riedl, J.: E-commerce recommendation applications. Data Mining and Knowledge Discovery **5**, 115–153 (2001)
57. Sinha, R., Swearingen, K.: Comparing recommendations made by online systems and friends. Proc. of the DELOS-NSF Workshop on Personalisation and Recommender Systems in Digital Libraries (2001)
58. Swearingen, K., Sinha, R.: Beyond algorithms: an HCI perspective on recommender systems. Proc. of SIGIR Workshop on Recommender Systems (2001)
59. Tang, W., Ma, Y., Chen, Z.: Managing trust in peer-to-peer networks. Journal of Digital Information Management **3**, 58–63 (2005)
60. Victor, P., De Cock, M., Cornelis, C., Pinheiro da Silva, P.: Towards a provenance-preserving trust model in agent networks. In: Proc. of Models of Trust for the Web WWW2006 Workshop (2006)
61. Victor, P., Cornelis, C., De Cock, M., Teredesai, A.M.: Key figure impact in trust-enhanced recommender systems. AI Communications **21**, 127–143 (2008)
62. Victor, P., Cornelis, C., De Cock, M., Pinheiro da Silva, P.: Gradual trust and distrust in recommender systems. Fuzzy Sets and Systems **160** 1367–1382 (2009)
63. Victor, P., Cornelis, C., De Cock, M., Teredesai, A.M.: A comparative analysis of trust-enhanced recommenders for controversial items. In: Proc. of the International AAI Conference on Weblogs and Social Media, pp. 342-345 (2009)
64. Victor, P., Cornelis, C., De Cock, M., Teredesai, A.M.: Trust- and distrust-based recommendations for controversial reviews. IEEE Intelligent Systems, in press.
65. Zadeh, L.A.: Fuzzy sets. Information and Control **8**, 338–353 (1965)
66. Zaihrayeu, I., Pinheiro da Silva, P., McGuinness, D.: IWTrust: Improving user trust in answers from the web. In: Proc. of the Third International Conference On Trust Management, pp. 384-392 (2005)
67. Zhang, S., Ouyang, Y., Ford, J., Makedon, F.: Analysis of a low-dimensional linear model under recommendation attacks. In: Proc. of the International ACM SIGIR Conference, pp. 517-524 (2006)
68. Ziegler, C., Lausen, G.: Propagation models for trust and distrust in social networks. Information System Frontiers **7**, 337–358 (2005)
69. Ziegler, C., Golbeck, J.: Investigating correlations of trust and interest similarity - Do birds of a feather really flock together?. Decision Support Systems **43**, 460–475 (2007)