Trust- and Distrust-Based Recommendations for Controversial Reviews

Patricia Victor, Chris Cornelis, Martine De Cock, Ankur M. Teredesai

Abstract

Recommender systems that incorporate a social trust network among their users have the potential to make more personalized recommendations compared to traditional collaborative filtering systems, provided they succeed in utilizing the additional trust and distrust information to their advantage. We compare the performance of several well-known trust-enhanced techniques for recommending controversial reviews from Epinions.com, and provide the first experimental study of using distrust in the recommendation process.
KEYWORDS: recommender system, trust network, distrust.

1 Introduction

Potential customers increasingly turn to the web to find information about the products they are interested in. Such information often comes in the form of online reviews. Nowadays, these reviews are not only written by experts anymore, but also by the customers themselves. In fact, user-supplied reviews are becoming more and more prevalent, think e.g. of the well-known e-commerce sites Amazon.com and Epinions.com, or the Internet Movie Database (imdb.com). Unfortunately, the wealth of information all too often makes it very difficult to find the reviews that will be truly helpful. A lot of systems try to alleviate this by computing one global score for the review, for example Amazon's 'x out of y people found the following review helpful'. Other applications generate the global score by combining techniques from the text classification area and opinion/sentiment analysis, see e.g. [1, 2]. However, a review that is helpful for one user is not necessarily equally useful for another user. This is e.g. reflected in Epinions' system, where members can evaluate the helpfulness of a review by assigning a rating which ranges from 'not helpful' (1/5) to 'most helpful' (5/5). If all the users who have read a particular review found it very helpful, it is reasonable to assume that a new user might appreciate it too. In such cases, a global score reflects the general agreement very well, and new users can immediately see that this is a review that they should read. However, the more challenging reviews are those that receive a variety of high and low scores, reflecting disagreement about the review. We call such reviews *controversial reviews* (CRs). More than in any other case, a helpfulness prediction for a user needs to be truly personalized when the review under consideration is controversial; i.e., when a review has both 'ardent supporters' and 'motivated adversaries', with no clear majority in either group.

This is where *recommender systems* (RSs) come into play. Such systems use information about their user's profiles and relationships to suggest items that might be of interest to them [3]. Recommender systems are often used to accurately estimate the degree to which a particular user (the target user) will like a particular item (the target item), and are hence particularly useful for predicting the helpfulness of CRs. For example, in Epinions, the ratings and relationships of the users are used to determine which reviews are shown to a particular user, and in what order.

Most widely used methods for making recommen-

dations are either content-based or collaborative filtering methods. Content-based methods suggest items similar to the ones that the user previously indicated a liking for. Hence, these methods tend to have their recommendation scope limited to the immediate neighbourhood of the users' past purchase history or rating record for items. RSs can be improved significantly by (additionally) using collaborative filtering (CF) [4], 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. The advanced recommendation techniques that we discuss in this paper adhere to the CF 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 item's content.

Research [5] has pointed out that people tend to rely more on recommendations from people they trust than on online RSs which generate recommendations based on anonymous people similar to them. This observation, combined with the growing popularity of open social networks and the trend to integrate e-commerce applications with RSs, has generated a rising interest in *trust-enhanced recommendation systems* (see e.g. [6, 7, 8, 9]). Such systems incorporate a trust network in which the users are connected by scores indicating how much they trust each other, and use that knowledge to generate 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 (trust propagation), etc.

Apart from trust, in a large group of users, each with their own intentions, tastes and opinions, it is only natural that also *distrust* begins to emerge. 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', which reflects the members that are distrusted by a particular user. The information in the WOT and block list is then used to make the ordered list of presented reviews more personalized. Other recent examples of web applications that work with negative evaluation concepts can e.g. be found in the political forum Essembly [10] or the technology news website Slashdot [11]. From a theoretical perspective, too, it is generally acknowledged that distrust can play an important role [9, 12, 13], but much ground remains to be covered in this domain.

This study provides a head-to-head comparison of the performance of several trust-enhanced algorithms in terms of their coverage and accuracy of recommendations for CRs, i.e., reviews that typically receive a variety of conflicting ratings. The comparison includes CF, as well as the well-known trust-enhanced strategies proposed by Golbeck et al. [6], Massa et al. [7], and O'Donovan et al. [8]. Furthermore, we study the effect of three new strategies to involve distrust into the recommendation process, viz. distrust as an indicator to reverse deviations, distrust as a filter, and distrust as a debugger of a WOT. We conduct our experiments on a large data set from Epinions; in the following section, we analyze its controversiality level. In Section 3, we discuss the rationale behind several well-known trust- and new distrust-enhanced algorithms, while their performance is analyzed in Section 4. To the best of our knowledge, the potential of utilizing distrust in the RS's process has not been experimentally evaluated before.

2 Controversial Reviews

Epinions.com is a popular e-commerce site where users can write reviews about products and assign a rating to them. Guha et al. [12] compiled a data set containing 1 560 144 reviews that received 25 170 637 ratings by 163 634 different users. These reviews are evaluated by assigning a helpfulness rating which ranges from 'not helpful' (1/5) to 'most helpful' (5/5). Most reviews receive very high scores, in fact, 76.9% of all ratings are 'most helpful'. This means that a simple algorithm that always predicts 5, or that uses the average score for the review as its prediction, will have a high accuracy. However, such recommendation strategies have difficulties coping with CRs. These reviews receive a variety of high

and low scores, reflecting disagreement about them.

A straightforward way to detect CRs in a data set is to inspect the standard deviation of the ratings for each review i (see e.g. [7]). The higher the standard deviation of the ratings for a review, the more controversial the review is. We denote this by $\sigma(i)$. A little under 10% of the reviews have a σ of at least 0.9; there are 103 495 such reviews in total. About 70% of all reviews have a σ that is lower than 0.5. This comes as no surprise, since the low values are due to the abundance of 5-ratings. However, standard deviation does not convey the full picture of controversiality, as we argued in [14]. To get a clearer picture of the true CRs, we introduced the following measure:

Definition 1 (Level of Disagreement) *For a system with discrete ratings on a scale from* 1 *to* M*, let* $\Delta \in \{1, \ldots, M\}$*. The* Δ*-level of disagreement for an item* i *is defined as*

$$(\alpha@\Delta)(i) = 1 - \max_{a \in \{1,\ldots,M-\Delta+1\}} \left(\frac{\sum_{k=a}^{a+\Delta-1} f_i(k)}{\sum_{k=1}^{M} f_i(k)} \right)$$

with $f_i(k)$ *the number of times that review* i *received rating* k*.*

This measure looks at how often adjacent scores appear w.r.t. the total number of received ratings. The underlying intuition is that different scores that are close to each other reflect less disagreement than different scores that are on opposite ends of the scale.

While a small σ typically entails a small level of disagreement, there is considerable variation for high values of σ (and vice versa) [14], which shows that σ and $\alpha@\Delta$ are significantly different measures that can be used together:

Definition 2 ($(\sigma^\star, \alpha^\star)$-controversial) *We call review* i $(\sigma^\star, \alpha^\star)$*-controversial iff* $\sigma(i) \geq \sigma^\star$ *and* $(\alpha@2)(i) \geq \alpha^\star$*.*

Applying this definition to the data set requires a parameter selection that is adapted to its characteristics, for example the predominance of rating value 5. We choose $\sigma^\star = 0.9$ and $\alpha^\star = 0.4$, obtaining a subset of 28 710 items for which a recommender system might experience high prediction difficulties. To ensure real controversiality, we further restrict the set to contain only the 1416 controversial reviews that have been rated at least 20 times, since the controversiality of reviews with few ratings may be due to chance.

3 Recommendation Strategies

RSs come in many flavours, including content-based, collaborative filtering and trust-based methods; the latter two being the ones most relevant to our current efforts.

3.1 Collaborative filtering

In collaborative filtering algorithms [4], 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. The classical CF-formula is given by (CF). The unknown rating $p_{a,i}$ for item i and target user a is predicted based on the mean $\overline{r}_a$ 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) [15]. 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^+.

$$p^{(1)}_{a,i} = \overline{r}_a + \frac{\sum_{u \in R^+} w_{a,u}(r_{u,i} - \overline{r}_u)}{\sum_{u \in R^+} w_{a,u}} \qquad \text{(CF)}$$

3.2 Trust-based methods

Trust-enhanced recommender systems often use information coming from a trust network in which users are connected by trust scores indicating how much they trust each other; in general, $t_{a,u}$ is a number between 0 and 1 indicating to what extent a trusts u.

Trust-based weighted mean refines the baseline strategy of simply computing the average rating for the target item. In particular, by including trust scores that reflect the degree to which the raters are trusted, it allows to differentiate between the sources; it is natural to assign more weight

to ratings of highly trusted users. The formula is given by (T1), in which R^T represents the set of users who evaluated i and for which $t_{a,u}$ exceeds a given threshold value. (T1) is at the heart of Golbeck et al.'s strategy using TidalTrust [6].

$$p_{a,i}^{(2)} = \frac{\sum_{u \in R^T} t_{a,u} r_{u,i}}{\sum_{u \in R^T} t_{a,u}} \tag{T1}$$

Another class of trust-enhanced systems is tied more closely to the CF algorithm. O'Donovan et al.'s *trust-based filtering* [8] adapts (CF) by only taking into account trustworthy neighbours, i.e., users in $R^{T+} = R^T \cap R^+$ instead of R^+. In other words, we only consider users who are trusted by the target user a and have a positive correlation with a:

$$p_{a,i}^{(3)} = \overline{r_a} + \frac{\sum_{u \in R^{T+}} w_{a,u}(r_{u,i} - \overline{r_u})}{\sum_{u \in R^{T+}} w_{a,u}} \tag{T2}$$

Instead of a PCC-based computation of the weights, one can also infer the weights through the relations of the target user in the trust network, as in (T1). We call this alternative for CF *trust-based CF*; see (T3) which adapts (T2) by replacing the PCC weights $w_{a,u}$ by the trust values $t_{a,u}$.

$$p_{a,i}^{(4)} = \overline{r_a} + \frac{\sum_{u \in R^T} t_{a,u}(r_{u,i} - \overline{r_u})}{\sum_{u \in R^T} t_{a,u}} \tag{T3}$$

This method is central to Massa et al.'s method with MoleTrust [7]. Note that, because the weights are not equal to the PCC, this procedure can produce out of bounds results. When this is the case, $p_{a,i}^{(4)}$ is rounded to the nearest possible rating.

A very important feature of trust-enhanced RSs is their use of *trust propagation operators*; mechanisms to estimate the trust transitively by computing how much trust an agent a has in another agent c, given the value of trust for a trusted third party b by a, and c by b. Both TidalTrust and MoleTrust invoke trust propagation to expand the set R^T of trusted users. However, the way they implement this operation differs significantly, see [6] and [7]. Although trust propagation is not used in (T2) because the trust scores are automatically generated [8], it is of course possible to do so; since trust scores are not used explicitly in this formula, we only need to specify how propagation enlarges the set R^T.

It has been demonstrated that including trust in the process significantly improves accuracy [6, 7]. On the other hand, the coverage of (T2) and (T3) remains lower than their classical counterpart (CF) [7]. One way of mending this is by using trust propagation. Another way is to maximize the synergy between CF and its trust-based variants, as done in the following algorithm [14]:

$$p_{a,i}^{(5)} = \overline{r_a} + \frac{\sum_{u \in R^T} t_{a,u}(r_{u,i} - \overline{r_u}) + \sum_{u \in R^+ \setminus R^T} w_{a,u}(r_{u,i} - \overline{r_u})}{\sum_{u \in R^T} t_{a,u} + \sum_{u \in R^+ \setminus R^T} w_{a,u}} \tag{T4}$$

The rationale behind this strategy is that we take into account all possible ways to obtain a positive weight for a user who has rated the target item, favouring a trust relation over a PCC-based one; in particular, if a user can be reached by a (in)direct trust relation, we use this value instead of the PCC to obtain the user's weight. In this way, we retain the accuracy benefit by first looking at the trusted users, while on the other hand the coverage can increase by taking into account neighbours for which no trust information is available.

3.3 Distrust-enhanced algorithms

It is generally acknowledged that apart from trust, also *distrust* can play an important role in trust networks, see e.g. [9, 12, 13, 16]. However, to the best of our knowledge, its potential for RSs has not been experimentally evaluated yet. This is due to several reasons, the most important ones being that very few data sets containing explicit distrust are available, and that there is no general consensus about how to propagate it and to use it for recommendation purposes.

In most current approaches, distrust information is modeled by means of [0,1]-valued scores $d_{a,u}$ that indicate to what extent a distrusts u, and that can be issued along with trust scores $t_{a,u}$. Moreover, various propagation strategies for (trust,distrust) couples have been presented [9, 16].

Distrust as a filter The use of distrust for RSs can be explored in several ways. A first strategy is to use distrust to filter out 'unwanted' individuals from collaborative recommendation processes. For instance, we propose

$$p_{a,i}^{(6)} = \overline{r_a} + \frac{\sum_{u \in R^+ \setminus R^D} w_{a,u}(r_{u,i} - \overline{r_u})}{\sum_{u \in R^+ \setminus R^D} w_{a,u}}, \qquad \text{(D2)}$$

in which (CF) is adapted so as to exclude distrusted users as neighbours; R^D represents the set of users who have rated the target item and that are distrusted by the target user to some degree. This is a similar approach to (T2) which restricts the neighbours to be trusted users.

Distrust as a debugger of a WOT In the same spirit, various researchers have suggested that distrust be used to debug a WOT (see e.g. [12, 13]): suppose that a trusts b completely, b fully trusts c and a completely distrusts c, then the latter fact invalidates the propagated trust result (viz. a trusts c). As such, distrust-enhanced algorithms may be useful to filter out 'false positives'. This strategy leads to two new formulas, (D1) and (D3), adaptations of (T1) and (T3) in which R^T is replaced by $R^T \setminus R^D$.

$$p_{a,i}^{(7)} = \frac{\sum_{u \in R^T \setminus R^D} t_{a,u} r_{u,i}}{\sum_{u \in R^T \setminus R^D} t_{a,u}} \qquad \text{(D1)}$$

$$p_{a,i}^{(8)} = \overline{r_a} + \frac{\sum_{u \in R^T \setminus R^D} t_{a,u}(r_{u,i} - \overline{r_u})}{\sum_{u \in R^T \setminus R^D} t_{a,u}} \qquad \text{(D3)}$$

We can also apply the above strategy to (T4), see (D4): we propose to use trust scores for those users which can be reached through propagation but for which no distrust propagation path can be found, and PCC scores for those in $R^+ \setminus R^{TD}$ with $R^{TD} = R^T \cup R^D$, i.e., the remaining ones which have a positive correlation with a but do not belong to R^T nor R^D, i.e. neither trust nor distrust information is available about them.

$$p_{a,i}^{(9)} = \overline{r_a} + \frac{\sum_{u \in R^T \setminus R^D} t_{a,u}(r_{u,i} - \overline{r_u}) + \sum_{u \in R^+ \setminus R^{TD}} w_{a,u}(r_{u,i} - \overline{r_u})}{\sum_{u \in R^T \setminus R^D} t_{a,u} + \sum_{u \in R^+ \setminus R^{TD}} w_{a,u}} \qquad \text{(D4)}$$

Distrust as an indicator to reverse deviations A third distrust strategy is the direct incorporation of distrust into the recommendation process by considering distrust scores as negative weights. We propose formula (D5), which is an extended version of (T3) in which distrust is regarded as an indication for reversing the deviation $r_{u,i} - \overline{r_u}$.

$$p_{a,i}^{(10)} = \overline{r_a} + \frac{\sum_{u \in R^T} t_{a,u}(r_{u,i} - \overline{r_u})}{\sum_{u \in R^T} t_{a,u}} - \frac{\sum_{v \in R^D} d_{a,v}(r_{v,i} - \overline{r_v})}{\sum_{v \in R^D} d_{a,v}} \qquad \text{(D5)}$$

If no distrusted users have rated the target item ($R^D = \emptyset$), the second fraction is omitted and (D5) collapses to (T3).

4 Comparative analysis

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. Users can evaluate other users by including them in their WOT (a list of reviewers whose reviews and ratings were consistently found to be valuable[1]), or by putting them in their block list (a list of authors whose reviews were consistently found to be offensive, inaccurate or low quality[1], thus indicating distrust). In our data set, these evaluations make up a WOT graph consisting of 131 829 users and 840 799 non self-referring (dis)trust relations. Note that the data set only contains binary (dis)trust values, hence in our experiments $t_{a,u}$ and $d_{a,u}$ in (T1)–(D5) can take on the values 0 (absence) and 1 (full presence) only.

To measure the performance of RSs, we work with the leave one out method. In particular, we use two

[1] See `www.epinions.com/help/faq/`

Table 1: Performance of trust-based algorithms

	ALGORITHM	% COV	MAE	RMSE
(CF)	CF with positive PCC	94	0.96	1.13
(T1)	Trust-based weight. mean	63	0.86	1.20
(T2)	Trust-based filtering	60	0.86	1.16
(T3)	Trust-based CF	63	0.87	1.16
(T4)	EnsembleTrustCF	94	0.94	1.11
(PT1)	Prop. Trust-based weight. mean	88	0.91	1.22
(PT2)	Prop. Trust-based filtering	84	0.94	1.13
(PT3)	Prop. Trust-based CF	88	0.99	1.16
(PT4)	Prop. EnsembleTrustCF	94	0.96	1.12

well-known accuracy measures, viz. mean absolute error (MAE) and root mean squared error (RMSE) [15]. Since reviews are rated on a scale from 1 to 5, the extreme values that MAE/RMSE can reach are 0 and 4. Besides accuracy, we also consider coverage: during the leave one out we count how many predictions can be generated for the hidden scores. We computed the coverage and accuracy of the algorithms discussed in Section 3, for the 1416 CRs described in Section 2.

4.1 Performance of trust-based approaches

Table 1 shows the relative coverage and accuracy for CRs in the data set. For simplicity, we only consider one-step propagation. For (PT1) and (PT3), we maintained the propagation strategy used in TidalTrust and MoleTrust respectively, while for (PT2) 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. For (PT4), we assign gradual propagated trust weights $t_{a,u} = (PCC + 1)/2$. In this way, users u who cannot be reached through a direct trust relation are still rewarded for their presence in a's propagated WOT.

Without propagation, it is clear that the coverage of (CF) and (T4) is superior to that of the other strategies, 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). Our new algorithm (T4) is most flexible, since having either some trust or a positive correlation is sufficient to make a prediction. On the other hand, (T2) 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. In between these extremes, the coverage for (T1) and (T3) is the same. This ranking of approaches in terms of coverage still applies when propagated trust information is taken into account, but note that the difference with CF has shrunk considerably. In particular, thanks to trust propagation, coverage increases with about 25%; except for (T4), for which the unpropagated version continues to score better than the propagated versions of (T1)–(T3).

Clearly, generating good predictions for CRs is hard. When focusing on the MAE, we notice that, without propagation, the trust-enhanced approaches all yield significantly better results than CF, which is in accordance with the observations made in [6, 7]. 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 (PT1)–(PT3) nears that of (CF) and (T4), so do the MAEs. However, the RMSEs give us a different picture: those of the trust-enhanced methods are generally higher than that of CF; recall that a higher RMSE means that more large prediction errors occur. One possible explanation is the fact that the set R^T of trusted acquaintances that have rated the target item is too small, 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.

We can also observe that our new algorithm is a valuable asset in the trust-enhanced domain: it achieves the best scores for CRs in terms of RMSE. The MAE for the unpropagated version of (T4) is higher than those of (T1)–(T3), but this is amply compensated by the much higher coverage. The coverage gap diminishes when taking into account propagation, but so does the difference in MAEs.

4.2 Utility of distrust

The results of our experiments can be found in Table 2. Let us first concentrate on the first line in which we evaluated the use of direct incorporation of distrust, see (D5). The high decrease in accuracy (increase of MAE as well as RMSE) compared to its trust-only counterpart (T3) is not compensated by a similar increase in coverage. This demonstrates that distrust should not be used as a way to reverse deviations.

The middle part of Table 2 focuses on the use of distrust to filter out 'unwanted' individuals without propagation. The results for (D2) and (D4) show that this kind of distrust filter has little or no effect on the results of (CF) and (T4): an item that is only rated by distrusted users is very uncommon because of the trust/distrust ratio in the data set (only 15% of all relations are distrust-based) and the fact that we are dealing with popular items. Hence, the unchanged relative coverage comes as no surprise. Also note that the accuracy slightly increases when filtering out distrusted users. Remark that we cannot use this kind of filter for (D1) and (D3) if no trust propagation is involved, since a user cannot put another user in his WOT and in his block list simultaneously.

In the lower part of Table 2 we further investigate the potential of filters; we focus on the utility of distrust as a debugger for a target user's web of trust. This results in extended versions of the strategies in the lower part of Table 1. When considering trust propagation and distrust filtering, we need a definition of distrust propagation. We adopt a binary approach in which the set of distrusted users of target user a contains all users who are directly distrusted by a, who are distrusted by the members of a's WOT, and users in the WOT of users who are distrusted by a. In other words, 'distrust your enemies' friends, as well as your friends' enemies'. This approach will probably unjustly exclude some users, but we consider it more important that it allows to filter out the greater part of the 'false positives' in the (propagated) WOT of the target user.

This strategy leads to a coverage decrease of about 2%-3% for (PD1), (PD3), and (PD4), compared to the original propagated (PT1), (PT3) and (PT4). In other words, using trust propagation (to reach more users) and distrust propagation (to filter out false positives) only has a marginal effect on the coverage. Debugging improves the performance of trust-based CF in terms of accuracy, but for the other two algorithms no clear conclusion can be drawn: the MAEs are never worse than their trust-only counterparts, but the RMSE results show discrepant values.

Table 2: Performance of distrust-based algorithms

	ALGORITHM	*% COV*	*MAE*	*RMSE*
(D5)	Trust+distrust based CF	67	0.97	1.31
(D2)	Trust+distrust-based filtering	94	0.95	1.12
(D4)	EnsembleTrust+Distrust CF	94	0.93	1.10
(PD1)	Prop. trust+distrust-based weight. mean	86	0.91	1.23
(PD2)	Prop. trust+distrust-based filtering	91	0.96	1.18
(PD3)	Prop. trust+distrust-based CF	86	0.93	1.14
(PD4)	Prop. EnsembleTrustCF	92	0.95	1.17

Remark that formula (PD2) differs from (PD1), (PD3) and (PD4) because it does not use trust propagation to enlarge the set of neighbours, but only distrust propagation to restrict the set. In this case, users that are directly or indirectly distrusted by the target user are filtered out. Hence the decrease in coverage compared to (CF), whereas for (D2) and (CF) the relative coverage remained unchanged. This strategy yields equal MAE's but increasing RMSEs.

Although the results presented in this section are still preliminary, they already indicate that regarding distrust as an indication to reverse deviations is not the line to take. Distrust as a filter and/or debugger looks more promising, but more experiments, on other data sets with different characteristics (for example with a higher distrust ratio), are needed to come to a more precise conclusion.

5 Conclusions

We have provided a comparative analysis of the performance of collaborative filtering and trust-enhanced recommendation algorithms for controver-

sial reviews (CRs). We have evaluated several well-known and new recommender techniques on a data set from Epinions which contains rating information, trust and distrust information. However, the data set has one shortcoming: the (dis)trust values are binary, making it impossible to investigate all aspects of the algorithms, since a lot of the existing trust-based approaches are based on the assumption that trust is a gradual concept. Unfortunately, no such data sets are publically available.

Trust-enhanced recommender systems experience difficulties when predicting ratings for CRs. A coverage and accuracy based comparison shows no clear winner among the three state-of-art trust-enhanced strategies proposed in [6, 7, 8]. We also provided the first experimental evaluation of the potential of distrust in RSs. To our knowledge, the data set we use is the only publicly available one that contains ratings and explicitly issued trust and distrust statements. Only 15% of all relations are distrust-based; consequently, experiments on future data sets with different characteristics may yield clearer answers to the question whether distrust can be used as a debugger and/or filter. The same remark also applies to other results in this paper. E.g., in data sets containing users with a more varying rating behaviour, more true CRs can be detected. It remains an open question whether distrust can play a beneficial role in recommender systems, but we believe that the reported observations and the questions raised along with them can help researchers to further examine its possibilities.

Acknowledgements

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

Biographies

Patricia Victor is a postdoctoral researcher at Ghent University. Her current research interests include social networks, trust modeling, and recommender systems. She received the Ph.D. degree in computer science from Ghent University in 2010. She can be reached at patricia.victor@ugent.be. Vakgroep WE02, Krijgslaan 281 (S9), 9000 Gent, Belgium.

Chris Cornelis is a postdoctoral researcher at Ghent University, supported by the Research Foundation–Flanders. His current research interests include the application of fuzzy and rough sets to recommender systems and machine learning. He received the Ph.D. degree in computer science from Ghent University in 2004. He can be reached at chris.cornelis@ugent.be. Vakgroep WE02, Krijgslaan 281 (S9), 9000 Gent, Belgium.

Martine De Cock is an associate professor at Ghent University (on leave) and a visiting associate professor at the Institute of Technology of the University of Washington, Tacoma. Her research interest is in computational web intelligence. She holds a M.Sc. and a Ph.D. degree in Computer Science from Ghent University (Belgium). From 1998 until 2005 her activities as a research assistant and a postdoctoral fellow were supported by the Fund for Scientific Research - Flanders. She worked as a visiting scholar in the BISC group at the University of California, Berkeley, and the Knowledge Systems Laboratory at Stanford University. She can be reached at mdecock@u.washington.edu and martine.decock@ugent.be. Box 358426 1900 Commerce St, Tacoma, WA 98402, USA; or Vakgroep WE02, Krijgslaan 281 (S9), 9000 Gent, Belgium.

Prof. Ankur M. Teredesai is an Associate Professor of Computer Science and Systems at the Institute of Technology, University of Washington, Tacoma. His research interests are data mining of networked data including pervasive and online social networks. He obtained a doctorate in Computer Science and Engineering from SUNY Buffalo for his work on Active Pattern Recognition using

Evolutionary Computing Techniques. He can be reached at ankurt@u.washington.edu. Box 358426 1900 Commerce St, Tacoma, WA 98402, USA.

References

[1] A. Ghose and P. Ipeirotis, Designing novel review ranking systems: predicting the usefulness and impact of reviews. Proceedings of the ninth international conference on Electronic commerce, ACM, 2007, p 303-310.

[2] Y. Liu, X. Huang, A. An, and X. Yu, Modeling and predicting the helpfulness of online reviews. Proceedings of the Eighth IEEE International Conference on Data Mining, IEEE Computer Society, 2008, p 443-452.

[3] G. Adomavicius and A. Tuzhilin, 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(6), 2005, p 734-749.

[4] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl, Grouplens: An open architecture for collaborative filtering of netnews. Proceedings of the 1994 ACM conference on Computer supported cooperative work, ACM, 1994, p 175-186.

[5] R. Sinha and K. Swearingen, Comparing recommendations made by online systems and friends. Proceedings of the Second DELOS Network of Excellence Workshop on Personalisation and Recommender Systems in Digital Libraries, The European Research Consortium for Informatics and Mathematics (ERCIM), 2001.

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

[7] P. Massa and P. Avesani, Trust metrics in recommender systems. In: Computing with Social Trust (Human-Computer Interaction Series) (ed. : J. Golbeck), Springer, 2009, p 259-285.

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

[9] P. Victor, C. Cornelis, M. De Cock, and P. Pinheiro da Silva, Gradual trust and distrust in recommender systems. Fuzzy Sets and Systems 160(10), 2009, p 1367-1382.

[10] T. Hogg, D. Wilkinson, G. Szabo, and M. Brzozowski, Multiple relationship types in online communities and social networks. Proceedings of the 2008 Spring Symposium on Social Information Processing, AAAI, 2008, p 30-35.

[11] J. Kunegis, A. Lommatzsch, and C. Bauckhage, The Slashdot Zoo: mining a social network with negative edges. Proceedings of the 18th International World Wide Web Conference, ACM, 2009, p 741-750.

[12] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, Propagation of trust and distrust, Proceedings of the 13th international conference on World Wide Web, ACM, 2004, p 403-412.

[13] C. Ziegler and G. Lausen, Propagation models for trust and distrust in social networks. Information System Frontiers 7(4-5), 2005, p 337-358.

[14] P. Victor, C. Cornelis, M. De Cock, and A.M. Terdesai, A comparative analysis of trust-enhanced recommenders for controversial items. Proceedings of the Third International Conference on Weblogs and Social Media, AAAI, 2009, p 342-345.

[15] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl, Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS) 22(1), 2004, p 5-53.

[16] A. Jøsang, S. Marsh, and S. Pope, Exploring different types of trust propagation. Lecture Notes in Computer Science 3986, Springer-Verlag, 2006, p 179-192.