# Trust- and Distrust-Based Recommendations for Controversial Reviews

**Patricia Victor and Chris Cornelis,** *Ghent University, Belgium*

**Martine De Cock and Ankur M. Teredesai,** *University of Washington, Tacoma*

*By comparing and extending several well-known trust-enhanced techniques for recommending controversial reviews from Epinions.com, the authors provide the first experimental study of using distrust in the recommendation process.*

Potential customers increasingly turn to the Web to find product information, which often comes in the form of online reviews. Nowadays, these reviews are written by other customers as well as experts. In fact, user-supplied reviews are becoming more prevalent, especially on well-known e-commerce sites such as Amazon.com, Epinions.com, and the Internet Movie Database (imdb.com).

Unfortunately, the wealth of information all too often makes it difficult to find truly helpful reviews. Many systems try to alleviate this by computing a global score for a 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 text classification and opinion/sentiment analysis.[1,2] In this study, however, we focus on techniques that produce a local, or personalized, helpfulness score.

In particular, we provide a head-to-head comparison of the performance of several trust-enhanced algorithms in terms of their coverage and accuracy of recommendations for controversial reviews (CRs)—reviews that typically receive a variety of conflicting ratings. The comparison includes collaborative filtering (CF) and the well-known trust-enhanced strategies proposed by Jennifer A. Golbeck,[3] Paolo Massa and Paolo Avesani,[4] and John O'Donovan and Barry Smyth.[5] Furthermore, we study the effect of three new strategies that include distrust in the recommendation process: distrust as an indicator to reverse deviations, distrust as a filter, and distrust as a web of trust (WOT) debugger. We conduct our experiments on a large data set from Epinions, a popular e-commerce site where users can write reviews about products and assign them a rating. We also discuss the rationale behind several well-known trust- and new distrust-enhanced algorithms, analyzing their performance. To the best of our knowledge, researchers have yet to experimentally evaluate the potential of utilizing distrust in the recommender system (RS) process.

## Online Reviews and Recommender Systems

A review that is helpful for one user is not necessarily equally useful for another.



1541-1672/11/$26.00 © 2011 IEEE
*Published by the IEEE Computer Society*

Epinions' review system reflects this, letting members evaluate a review's helpfulness by assigning a rating that ranges from "not helpful" (1/5) to "most helpful" (5/5). If all the users who 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, and new users can immediately see that this is a review that they should read. However, CRs that receive both high and low scores are more challenging. More than in any other case, a helpfulness prediction for a user needs to be truly personalized when the review under consideration is controversial—that is, when a review has both ardent supporters and motivated adversaries, with no clear majority in either group.

This is where RSs come into play. Such systems use information from users' profiles and relationships to suggest possible items of interest.[6] They help 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, Epinions utilizes user ratings and relationships to determine which reviews are shown to a particular user, and in what order.

Most widely used methods for making recommendations are either content-based or CF methods. Content-based methods suggest items similar to the ones that the user previously purchased or reviewed favorably. Hence, the scope of these recommendations is limited to the immediate neighborhood of the users' past purchase history or ratings. By identifying users with tastes similar to the target user (neighbors) and by computing predictions based on the ratings of these neighbors, CF can significantly improve RSs.[7] The advanced recommendation techniques that we discuss in this work adhere to the CF paradigm, in the sense that a recommendation for a target item is based on other users' ratings of that item rather than on an analysis of the item's content.

Research indicates 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.[8] 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 RSs.[3–5,9] Such systems incorporate a trust network in which the users are connected by scores that indicate how much they trust each other. They then use that knowledge to generate recommendations; that is, users receive recommendations for items rated highly by people in their WOT or even by people who are trusted by those in their WOT (trust propagation), and so on.

In a large group of users, each with their own motivations, tastes, and opinions, it is only natural that distrust begins to emerge. For example, Epinions first allowed users to include other members in a personal WOT (based on their quality as a reviewer), but it later also introduced the concept of a personal *block list*, which consists of members the user distrusts. Epinions then uses the WOT and block list information to personalize the ordered list of presented reviews.

Other recent examples of Web applications that work with negative-evaluation concepts include the political forum Essembly[10] and the technology news website Slashdot.[11] Also from a theoretical perspective, distrust can play an important role,[9,12,13] but much ground remains to be covered in this domain.

## Controversial Reviews

R. Guha and his colleagues compiled a data set containing 1,560,144 Epinions reviews that received 25,170,637 ratings by 163,634 different users.[12] Most reviews received very high scores; in fact, 76.9 percent of all ratings were "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.

A straightforward way to detect CRs in a data set is to inspect the standard deviation of the ratings for each review $i$.[4] The higher the standard deviation for a review's ratings, the more controversial the review. We denote this by $\sigma(i)$. A little under 10 percent of the reviews (103,495 in total) have a $\sigma$ standard deviation of at least 0.9. Approximately 70 percent of all reviews have a $\sigma$ 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 an earlier work.[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 $\Delta$ level of disagreement for an item $i$ is defined as

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

where $f_i(k)$ is the number of times that review $i$ received rating $k$.

This measure looks at how often adjacent scores appear with regard to 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. Although a small $\sigma$ typically entails a small level of disagreement, there is considerable variation for high values of $\sigma$ (and vice versa),[14] which shows that $\sigma$ and $\alpha@\Delta$ are significantly different measures that we can use together:

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

Applying this definition to the data set requires a parameter selection that is adapted to its characteristics—for example, the predominance of the rating value 5. We choose $\sigma^* = 0.9$ and $\alpha^* = 0.4$, obtaining a subset of 28,710 items for which a recommender system might experience high prediction difficulties. We further restricted the set to contain only the 1,416 CRs that have been rated at least 20 times because those with only a few ratings might be due to chance.

## Recommendation Strategies

RSs come in many flavors, including content-based, CF, and trust-based methods. The latter two are the ones most relevant to our current efforts.

### Collaborative Filtering

CF algorithms[7] can predict a rating of target item $i$ for target user $a$ using a combination of the ratings of the neighbors of $a$ (similar users) that are already familiar with item $i$. The classical CF formula is

$$p_{a,i}^{(1)} = \bar{r}_a + \frac{\sum_{u \in R^+} w_{a,u}(r_{u,i} - \bar{r}_u)}{\sum_{u \in R^+} w_{a,u}} \quad (1)$$

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 $i$. Equation 1 also accounts for 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 $R^+$.

### Trust-Based Methods

Trust-enhanced RSs 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. By including trust scores that reflect the degree to which the raters are trusted, it lets us differentiate between the sources; it is natural to assign more weight to ratings of highly trusted users:

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

where $R^T$ represents the set of users who evaluated $i$ and for which $t_{a,u}$ exceeds a given threshold value. Equation 2 is at the heart of Golbeck's strategy using TidalTrust.[3]

Another class of trust-enhanced systems is tied more closely to the CF algorithm. O'Donovan and Smyth's trust-based filtering adapts Equation 1 by only taking into account trustworthy neighbors—that is, users in $R^{T+} = R^T \cap R^+$ instead of $R^+$.[5] In other words, we only consider users who are trusted by the target user $a$ and have a positive correlation with $a$:

$$p_{a,i}^{(3)} = \bar{r}_a + \frac{\sum_{u \in R^{T+}} w_{a,u}(r_{u,i} - \bar{r}_u)}{\sum_{u \in R^{T+}} w_{a,u}} \quad (3)$$

Instead of a PCC-based computation of the weights, we can also infer the weights through the relations of the target user in the trust network, as in Equation 2. We call this alternative *trust-based CF*. For example, this formula,

$$p_{a,i}^{(4)} = \bar{r}_a + \frac{\sum_{u \in R^T} t_{a,u}(r_{u,i} - \bar{r}_u)}{\sum_{u \in R^T} t_{a,u}} \quad (4)$$

adapts Equation 3 by replacing the PCC weights $w_{a,u}$ with the trust values $t_{a,u}$. This method is central to Massa and Avesani's method with MoleTrust.[4] Yet, 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.

An important feature of trust-enhanced RSs is their use of *trust propagation operators*, which are 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 $a$ as well as $c$ by $b$. Both TidalTrust and MoleTrust invoke trust propagation to expand the set $R^T$ of trusted users. However, they implement this operation in significantly different ways.[3,4] Although Equation 3 does not use trust propagation because the trust scores are automatically generated,[5] it is of course possible to do so. Because this formula does not explicitly use trust scores, we only need to specify how propagation enlarges the set $R^T$.

Previous research demonstrated that including trust in the process significantly improves accuracy.[3,4]