Utility of Distrust in Online Recommender Systems

Capstone Project Report

Uma Nalluri

Computing & Software Systems
Institute of Technology
Univ. of Washington, Tacoma
unalluri@u.washington.edu

Committee: Ankur Teredesai (chair), Martine De Cock
Advisor: Patricia Victor

Abstract

Online Recommender Systems (RS) utilize readily available user profiles and preferences to recommend items that the user might be interested in. Many researchers have been investigating how trust among agents, both human and computer, can be factored into recommender systems to improve the number and quality of predictions. Although there is a fair amount of focus on hypothesizing how distrust may help with improving predictions, there has not been any empirical evidence to prove or disprove these hypotheses. This paper is a novel attempt to evaluate the utility of rating-prediction models using distrust and compares the results with existing trust-only-based models. The experiments are conducted using Epinions.com datasets downloaded from Trustlet.org. The results of the experiments suggest that the accuracy and coverage of the recommendations can be improved by using distrust.

1. Introduction

This paper discusses the empirical evaluation of distrust in online recommendation systems. Traditional Recommendation Systems (RS) widely use a Collaborative Filtering (CF) technique to predict ratings. With a CF technique, product ratings are predicted for the target user by identifying users that have similar purchase histories. The profiles are compared by matching the items in the purchased histories to find similar users. For example, amazon.com presents recommendations based on user's purchase history – "user who bought x also bought y". The limitation with CF is that the number of predictions is limited by the number of similar users found in the network. This problem is more for cold start users that are new and have very little history. This problem is addressed to some extent by extending CF with trust. Here users designate explicit trust in other user's reviews. The set of trusted users are used to predict the ratings. Massa et al. [5] and O'Donovan [3] discuss trust-enhanced models and the results of their experiments.

In order to evaluate distrust, we propose prediction models for RS using both trust and distrust. Review ratings are predicted for each of the proposed models using the Extended Epinions Dataset. The Extended Epinions Dataset is available on trustlet.org. It comes with user review ratings as well as the explicit trust and distrust designations among the users.

The proposed trust-with-distrust models are extensions of trust-prediction models proposed by Massa et al [5] and Golbeck [13]. The trust values from the Epinions dataset are propagated using Massa's trust propagation model. Ratings are predicted using both explicit trust as well as propagated trust. These results from trust-only models are compared with the results from the proposed trust-with-distrust models.

Massa's propagation model is based on the transitive nature of trust. As depicted in Figure 1, if A Trusts B, and B Trusts C, could we predict how much A Trusts C?

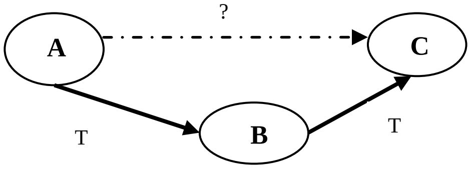

Figure 1 –Transitive trust

Massa's propagation model as described in Equation (1) is a linear model. As the number of hops ($s_{a,u}$) from the source User 'a' to the user in the network 'u' increases, the trust value decreases [5]. The maximum allowed distance from the source user to the user in the trust network is also referred to as horizon, i.e. $s_{a,u} = h + 1$, then the inferred trust $t_{a,u}$ becomes 0.

$$\boldsymbol{t_{a,u}} = {}^{\boldsymbol{(h - S_{a,u} + 1)}}\!\big/_{\boldsymbol{h}} \qquad \textbf{(1)}$$

$t_{a,u}$ = *trust value of user 'a' in user 'u'.*
h = *Maximum propagation path (constant).*
$s_{a,u}$ = *Distance between the source user and the user at the horizon.*

The Epinions dataset has only 123,705 distrust relations as opposed to 717,667 trust relations. In order to increase the number of distrust statements we propose a distrust-propagation technique we term *propagation of trust-with-distrust* thereby extending Massa's trust-propagation model [5].

Extending Massa's trust-propagation model [5] to distrust is complex because we cannot consider distrust as transitive as we did in case of trust. For example, in Figure 1, if User A Distrusts User B, and B Trusts C then we cannot definitively say whether A Trusts C or A Distrusts C. Taking the limitation of distrust into consideration, we have extended the propagation model of trust-with-distrust. This model is further described in Section 4.

Even though Ziegler [12] and Guha [13] have proposed a few hypotheses on distrust-propagation and computation models for distrust, until now there has been no effort to provide empirical evaluation of distrust-enhanced prediction models for RS. In this paper we attempt such an empirical evaluation of distrust in RS, and demonstrate that distrust does indeed improve prediction accuracy and coverage.

2. Trust-enhanced Recommedation Systems

Early RS gathered data through surveys to create user categories and use machine-learning classifiers to classify users into categories. Recommendations are generated based on these classifications [1]. Among the classical RS techniques, Content-Based and Collaborative Filtering are the most commonly used. Content-Based RS match the user interests indicated in their profiles to the item descriptions. This requires that the system be aware of the content information of the item before the recommendations are generated [3]. An example of the Content-Based RS is a movie recommender system where content information like genre, language, actors, directors, and producer is matched against the learned preferences of the user in order to select a set of appropriate movie recommendations. For example, if a customer of a video rental store has rented only comedies, the system may only recommend comedies and not take the other possible user interests into consideration.

The second technique is Collaborative Filtering (CF). CF identifies users whose tastes are similar to those of the target user and generates recommendations. The typical user profile in a RS contains a product and rating vector. Traditional CF computes the similarities between profiles by using Pearson's Correlation Coefficient. For each target user the rating for an item is predicted in combination with each individual partner. These individual predictions are aggregated by factoring in the contribution of a partner's prediction according to its degree of similarity with the target users. This ensures that the more the similarity between the users the larger the impact on the final rating prediction will be [5]. While the primary goal of a RS is to increase the number of recommendations for a given user, with CF this number depends on the number of similar users found for the target user and also on the number of items those users have rated. However, not every user rates every item that is in the system. This phenomena where similar users typically rate only a limited number of items is known as sparseness. This is even worse for new users (also referred to as cold start users), who have not rated many items yet.

Researchers have shown that extending CF with trust values can help increase coverage and accuracy of recommendations [5][3][13]. Coverage implies the number of items for which the RS can predict the ratings [6]. It is computed as the ratio of the amount of items for which the RS can predict the ratings versus the total amount of items available to the RS. A commonly used approach to measure accuracy is leave-one-out technique. Leave-one-out technique involves hiding a known rating and predicting it. The predicted rating is compared with the actual rating and the difference is the predicted error. Averaging this error over all the predictions gives the Mean Absolute Error (MAE).

Most online acquaintances' do not know each other before meeting online. Hence, most social network sites allow users to designate their trust in other users. A user's explicit trust designation for other users in the network is known as the user's personal web of trust [2]. Epinions.com is an e-commerce site where users can designate their trust or block users whose reviews are not helpful. In this paper we referred to blocking as distrust. Massa et al extended CF with trust in predicting the rating of a particular item for the target user [5]. In his approach trust was propagated to address the data sparseness.

Our evaluation of trust-with-distrust algorithms is the extensions of the ideas discussed in [14]. The main focus of [14] is to find ways to interpret and incorporate distrust in RS in such way that improves the quality of predictions. Do we get better results when distrust is ignored or when distrust is used to reverse recommendations. The goal of [14] is also to compare trust-based and distrust-based algorithms on controversial items (items with good and bad ratings) from Epinions.com dataset. The authors used a leave-one-out technique to evaluate the prediction ratings. Accuracy of the prediction model is measured by finding the difference between the actual ratings and the prediction rating. Using the prediction error for each rating predicted, Mean Absolute Error (MAE) and Root Mean squared error (RMSE) are computed. Along with accuracy they also measured the coverage i.e. for some items the number of users for which the prediction can be made. In the conclusion they said none of the algorithms stands out among others.

3. Trust and Distrust Propagation

Figure 2 depicts an example trust propagation scenario. Maximum propagation distance is 3 in this example. A →B and A → C are the explicit trust User A has in User B and C, the trust value being 1. This is also denoted as Level 1. To propagate this trust to Level 2, the web of trust for users that are at Level 1 is considered. In this case the Level 1 users are B and C. B Trusts A. But, A is the source user, resulting in self-trust. Hence, this trust relationship has been eliminated. Elimination of trust relationships from the trust propagation network are shown with a crossed out arrow in Figure 2. Level 1 User C Trusts E and B. In this scenario, from A there are two paths to B, namely A → B and A → C → B. In case where there is more than one path to a user from the source, only the shortest path to a user from the source, only the shortest path is considered. In this case the shortest path from

A to B is A → B. This eliminates the trust relationship between C and B from the propagation network. Next, User A Trusts C and C Trusts E. So, using Equation (1) user A's Trust in user E can be predicted. The predicted trust value is 0.66. The derived trust relation A → E is depicted with a dotted line in Figure 2. User E is at Level 2. To propagate the trust further to Level 3, the web of trust for all the Level 2 users is considered. In this case there is only User E at Level 2. User E Trusts User F. Using Equation (1) User A's Trust in User F can be predicted. The predicted value is 0.33. Before propagation, User A had only two trust relations. After propagation User A has four trust relations. Hence, trust propagation increases the number of trust relations for each user, thereby increasing the number of users that contribute to each prediction. We expect this to improve the prediction coverage.

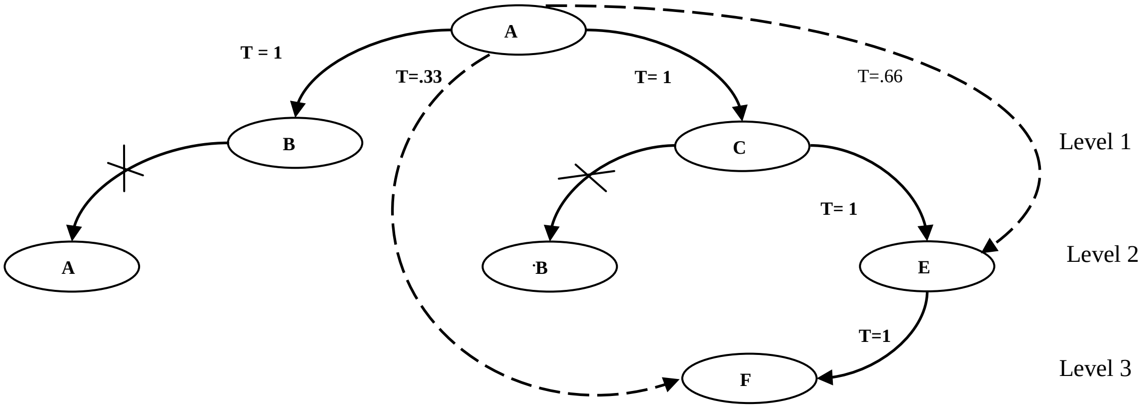

Figure 2 - Propagation of Trust

Increasing the number of trust relationships also helps with the accuracy of the predicted rating because the ratings are predicted by aggregating the ratings of all the trusted users. In case of CF the ratings are computed by aggregating the ratings of similar users. Similarities can be computed only with users who had purchased similar items. With explicit trust designations and further computing the propagated trust we can increase the number of user ratings that would participate in the prediction. We expect this to improve the prediction accuracy.

How to model distrust in online social networks is complex. Conceptually and intuitively, trust and distrust are different. Guha [14] describes how distrust can be considered a negative trust. For example, if Jane distrusts Jake, we can interpret that Jane's ratings negatively correlate to Jake's ratings. However, Guha also points out that negative trust doesn't cancel out trust. i.e., if Jane distrusts Jake and Jake distrusts Linda, this doesn't imply Jane trusts Jake. Unlike trust, propagation of distrust is not transitive. For instance, we think generalizing that enemy of an enemy is a friend is not applicable in this context.

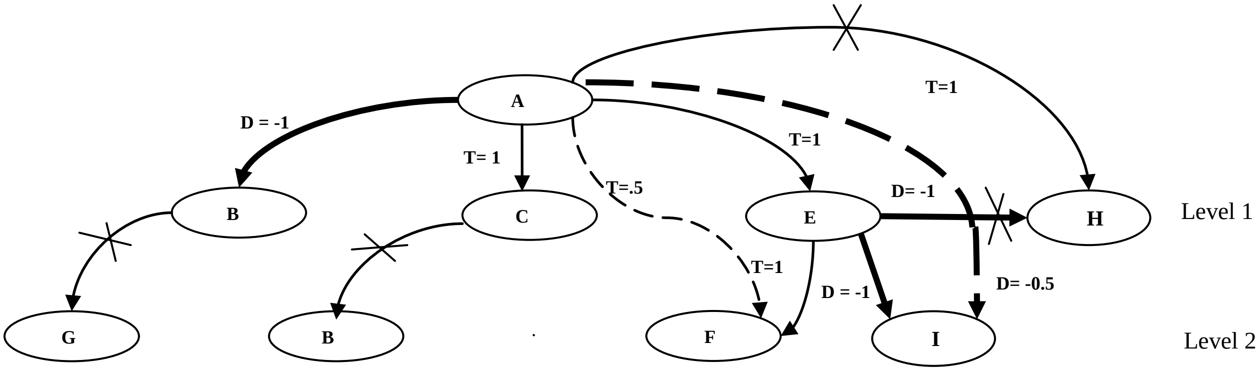

Figure 3 – Propagation of Trust-with-Distrust

In this experiment we have taken the trust and distrust relationships from the extended Epinions dataset and propagated trust and distrust values by modifying Massa's trust propagation [5]. We set a maximum propagation distance/horizon as '2'. Propagation of trust-with-distrust is performed in four steps. The first step is propagation of trust and distrust by using Massa's propagation represented in Equation (1). In order to make a distinction between trust and distrust values, distrust has been represented as negative value. Figure 3 represents distrust relationships between users A → B, E → I, E→ H, and A → I. The distrust value is represented as negative value. Distrust is propagated just as trust i.e. the absolute value of the computed trust and distrust for users at the same level is always equals. However, distrust values are represented as negative values. In Figure 3 the inferred distrust of User A on User I is computed as 0.5. Because I is distrusted by E the inferred trust between A and I is represented as -0.5. The second step is to identify all possible paths from the source user to each user in the network. If the source has multiple paths to another user in the propagation network, we have considered the shortest path and eliminated the rest. As depicted in Figure 3, User A can reach B directly or through C. We consider the shortest path namely A → B and eliminated A → C → B. The third step is to identify all the users that the source user shares a distrust relationship with and eliminate the web of trust of each of the distrusted users from the network. Please note that we are not eliminating the distrust relationship that the source shares with this user rather we are ignoring the other relationships that this distrusted user has. As depicted in Figure 3, the propagation is stopped at user B because user A Distrusts User B. Hence, we have eliminated User B's web of trust namely the relationship that B Trusts G. This is represented by crossing the relationship arrow. The fourth step is to identify all the users in the propagation network that are trusted by some users and distrusted by others. We eliminated such users. As depicted in Figure 3, User H is trusted by A and distrusted by user E. Hence, we have eliminated user H from the propagation network. This is represented by crossing the relationship arrow in Figure 3.Since we have eliminated these users, we wanted to see if considering distrust with trust would help with prediction error or not. The results of our experiments demonstrate that this elimination does reduce the error.

4. Data

Epinions.com is a popular e-commerce site where users can rate products, write product reviews and assign ratings to the reviews. The extended Epinions dataset is readily available on Trustlet.org. This dataset has both trust and distrust relationships. We took the data that has user ratings on the reviews of other users and user's explicit trust/distrust relationships. There are a total of 13,668,320 review ratings. Reviews are evaluated by assigning a helpfulness rating which ranges from 'not helpful' (1/5) to 'most helpful' (5/5). There are a total of 841,372 trust ratings. There are 573 ratings where users trust themselves. After eliminating the self trust ratings, we end up with 840,799 of which the total number of trust ratings is 717,129 and the total number of distrust ratings is 123,670. Note that we do not have information on consumer products and product ratings, but work with reviews and review ratings instead; in other words, we evaluate a 'review recommender system'. Hence, in this context, an item denotes a review of consumer goods. [7]

5. Proposed Prediction Models

Massa's prediction model as described in Equation (2) predicts the rating of user 'a' for item 'i' by aggregating all the trusted users that have also rated item 'i'. $\boldsymbol{t_{a,u}}$ is the trust value of user 'a' on user 'u' in the propagated-trust network.

$$\boldsymbol{p_{a,i} = \overline{r_a} + \frac{\sum_{u=1}^{k} t_{a,u}(\overline{r_u} - r_{u,i})}{\sum_{u=1}^{k} t_{a,\mathbf{u}}}} \tag{1}$$

$t_{a,u} = (Propagated) trust\ value$
$P_{a,i} = Predicted\ score\ of\ item\ 'i' for\ user\ 'a'.$
$\overline{r_u} =$
$Mean\ rating\ by\ user\ 'a'\ after\ removing\ user\ 'a' rating\ for\ item\ 'i'$
$r_{u,i} = Score\ of\ item\ 'i' given\ by\ user\ 'a'.$

For our experiments we propose to use the trust-with-distrust models represented as Equation (3) , Equation (4), and Equation (5) . These are prediction models that use trust and distrust. They are extensions of Massa's trust-propagation model represented as Equation (2). Here distrust is considered as reverse recommendation.

In Equation (3) if the user is distrusted, the influence of the distrusted user's rating is subtracted from the aggregate.

$$\boldsymbol{p_{a,i} = \overline{r_a} + \frac{\sum_{u=1}^{k}(t_{a,u} - d_{a,u})(\overline{r_u} - r_{u,i})}{\sum_{u=1}^{k}(t_{a,u} - d_{a,u})}} \tag{3}$$

$d_{a,u} = (Propagated)\ distrust\ value$

In Equation (4) the idea of reverse recommendation is implemented by computing the aggregate value of the influence of trusted user's rating of review 'i' separately from the aggregate value of the influence of distrusted user's rating of review 'i'. The aggregate of distrusted users is subtracted from the aggregate of trusted users.

$$\boldsymbol{p_{a,i} = \overline{r_a} + \frac{\sum_{u=1}^{k} t_{a,u}(r_{u,i} - \overline{r_u})}{\sum_{u=1}^{k} t_{a,u}} - \frac{\sum_{u=1}^{k}\ d_{a,u}(r_{u,i} - \overline{r_u})}{\sum_{u=1}^{k} d_{a,u}}} \tag{4}$$

Equation (5) is quite similar to Equation (4). But we are taking an average of trust and distrust factor. This is to see if averaging these two factors may improve the predictions.

$$\boldsymbol{p_{a,i} = \overline{r_a} + \frac{\sum_{u=1}^{k} t_{a,u}(r_{u,i} - \overline{r_u})}{2\sum_{u=1}^{k} t_{a,u}} - \frac{\sum_{u=1}^{k}\ d_{a,u}(r_{u,i} - \overline{r_u})}{2\sum_{u=1}^{k} d_{a,u}}} \tag{5}$$

Equation (6) is an extension of Golbeck's prediction model [11]. Trust factor is the aggregation of product of trust value and the trusted user's review rating. For each distrusted user, the difference between the maximum possible rating 5 as distrust value. Distrust factor is the aggregation of distrust value and the distrusted user's rating.

$$\boldsymbol{p_{a,i} = \frac{\sum_{u=1}^{k} t_{a,u} r_{u,i}}{\sum_{u=1}^{k} t_{a,u}} - \frac{\sum_{u=1}^{k}\ (5 - d_{a,u}) r_{u,i}}{\sum_{u=1}^{k} d_{a,u}}} \tag{6}$$

We have tested the validity of these algorithms for a subset of the Epinions Dataset users. The results are discussed in section 6.

6. Experiments and results

The Extended Epinions dataset has 13,572,033 user-review rating pairs. These are known ratings for which we can predict the ratings and compute MAE to verify our models. After trust propagation some users have as many as 14,000 trust relationships, which means for each prediction the aggregation involves participation of up to 14,000 ratings. Our initial experiments using the entire dataset, 13,572,033 user, review rating pairs, resulted with very high algorithm runtime. Hence, the experiments were conducted for only the top 1000 users that have the largest number of trust and distrust relationships as a proof of concept.

SQL Server 2005 was used for the data layer. Data is modeled with indexes and materialized views for improved access time. The predictions were performed in two stages. First, the propagated trust ($t_{a,u}$) and trust with distrust ($t_{a,\mathrm{u}}$,$d_{a,u}$) networks were computed. These values are stored in a database table. Later, indexes were created on this table to improve the access time for computing the predicted ratings. Stored procedures were created for predictions, compiled and stored at the database level. On compilation optimized query plans are created and stored to improve performance.

We ran Massa's trust propagation as well as the trust-with-distrust propagation algorithm described in Section 3 on the entire dataset for a maximum horizon of 2. We took a sample of top 1000 users with most trust and/or distrust relationships from each network for our experiments. From here on we refer to the top 1000 users with most relationships as sample test users.

In order to re-validate Massa's empirical evidence i.e. to show that trust propagation improves the MAE of predictions, we conducted two experiments. First, we predicted the review ratings for the sample test users from the explicit trust network. Resultant MAE for these users is 0.045. Second, we predicted the ratings for the sample test users from the trust propagation network. The MAE with trust propagation is 0.018. This clearly shows that trust propagation improves the prediction error. This also confirms the correctness of our algorithm implementation.

Next, we conducted experiments on the sample test users from trust-with-distrust propagation network. Review ratings were predicted for the sample test users by implementing algorithms represented in Equations (3), (4), (5), and (6). The MAE and the number of predictions are compared with the trust-only propagation results. Table 1 shows the MAE and coverage. We did not get any decreased error with Equations (4), (5) and (6). But, Table 1 clearly shows that Equation 3 (trust-with- distrust propagation) results in better MAE (0.0045) than Equation (2) (trust-only propagation) MAE (0.018). However, the coverage with Equation (3)(4, 829,037) did go down compared to coverage with Equation (2) (4,933,322). This can be attributed to those eliminated users that are trusted and distrusted by the source user.

Algorithms	**Direct Trust Equation (2)**	**Trust Propagation Equation (2)**	**Trust with Distrust Equation (3)**	**Trust with Distrust Equation (4)**	**Trust with Distrust Equation (5)**	**Trust with Distrust Equation (6)**
Mean Absolute Error	0.045	0.018	**0.0045**	0.029	0.22	0.72
Number of Predictions	9,870,686	4,933,322	**4,829,037**	4,825,757	4,825,757	4,825,757

Table 1 – Results of the experiments

7. Implementation Issues

Our goal was to propagate trust-only data and run the algorithm specified in Equation (2). Propagate trust-with-distrust for the same maximum propagation distance. Then use the propagated trust-with-distrust data to run the proposed algorithms in Equation (3), Equation (4), Equation (5), and Equation (6). The size of the data in terms of the number of review ratings for each user and the trust and distrust relationships for each user in the Extended Epinions.com dataset is very large because of which we had to experiment with different approaches to implement the algorithms before we accomplished a reasonable run time for each algorithm.

Our initial implementation using in-memory data structures for computation has resulted in out-of-memory issues. Next, we attempted using SQL Server 2005 for storing and retrieving data. On SQL Server we have created tables, clustered views and indexes to improve the data access time. The algorithms are re-implemented in VB.NET and with ADO.NET calls to SQL Server 2005. Computation is distributed between the .NET program and SQL Server stored procedures for the best possible run-time. During the initial execution of the propagation algorithm we encountered SQL query timeout issues. This issue was resolved by adjusting the SQL Server time out parameters appropriately. To increase the memory and processing capacities, we have created a client/server environment, SQL Server 2005 was on a different machine acting as a server and .NET program was run on a client machine. The initial trust-propagation algorithm run against the entire sample took three days for a maximum propagation path of four of Equation (1). To implement the proposed trust-with-distrust propagation for the same maximum propagation distance, four, involved more number of trust relationships as well as additional computation. For example, we not only had to identify the users that have more than one path from the source user but also had to keep the shortest path and eliminate the rest. Of which we only keep the user with the shortest path. In addition, we also have to eliminate the users that are trusted by one or more users but also distrusted by others. This algorithm did not finish by the 5th day.

Because, we were running short of time we had to come up with more manageable sample of data that best fits our experiments. We reduced the maximum propagation distance from 4 to 2. We ran both trust-only propagation [5] and proposed trust-with-distrust propagation algorithms. We selected the top 1000 users that have the most number of trust relationships from both trust-only and trust-with-distrust propagation results. Next, we implemented the algorithms described in Equation (2), Equation (3), Equation (4), Equation (5), and Equation (6) were implemented. The algorithms are run on the top 1000 connected users.

8. Discussion and future research

The proof of concept confirms that we get better prediction accuracy with trust-with-distrust prediction model described in Equation (3) in comparison with trust-only prediction model described in Equation (2). This can be attributed to the proposed trust-with-distrust propagation model. Using this model we identify those users that are trusted by some but also distrusted by others. Such users can have negative effect on the predictions. By propagating trust-with- distrust we can identify and eliminate such users thus improving the accuracy. At the same time, eliminating some of these users reduced the number of recommendations. This is clearly because the number of users that participated in the prediction are less in case of trust-with-distrust when compared with trust-propagation.

We have used a linear propagation model for this experiment. Future research can use a different propagation model such as probabilistic model to refine the trust network to obtain better results.

References

[1] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," User Modeling and User-Adapted Interaction, vol. 12, p. 39, 2002.

[2] J. Goldbeck, "Computing and applying trust in web-based social netowrks," in Computer Science. vol. Ph.D College Park: University of Maryland, 2005.

[3] J. O'Donovan and B. Smyth, "Trust in recommender systems," in Proceedings of the 10th international conference on Intelligent user interfaces San Diego, California, USA, 2005, pp. 167-174.

[4] J. L. Herlocker, J. A. Konstan, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," ACM Trans. Inf. Syst., vol. 22, p. 48, 2004.

[5] P. Massa and P. Avesani, "Trust-aware collaborative filtering for recommender systems," In Proc. of Federated Int. Conference On The Move to Meaningful Internet, vol. Lecture notes in computer science, p. 16, 2004.

[6] P. Victor, C. Cornelis, M. D. Cock, and A. Teredesai, "Key Figure Impact in Trust-Enhanced Recommender Systems."

[7] P. Victor, C. Cornelis, M. D. Cock, and P. P. d. Silva, "Gradual Trust and Distrust in Recommender Systems," To appear in Fuzzy Sets and Systems.

[8] J. Golbeck and J. Hendler, "Reputation Network Analysis for Email Filtering," in Proc. of the Conference on Email and Anti-Spam (CEAS), Mountain View, CA, 2004.

[9] D. Artz, and Y. Gil, "A survey of trust in computer science and the Semantic Web," Web Semant., vol. 5, no. 2, 2007, pp. 14.

[10] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust," ACM Press In WWW '04: Proceedings of the 13th international conference on World Wide web New York, NY, USA, 2004, pp. 403-412.

[11] J. Golbeck, "Filmtrust: movie recommendations from semantic web-based social networks," in Consumer Communications and Networking Conference. vol. 2, 2006, pp. 1314- 1315.

[12] C. Ziegler, and G. Lausen, "Propagation Models for Trust and Distrust in Social Networks," Information Systems Frontiers 7. vol. 4-5, 2005, pp. 337-358.

[13] R. Guha, "Open rating systems," Stanford Knowledge Systems Laboratory, Stanford, CA, USA, 2003.

[14] P. Victor, C. Cornelis, M. De Cock, and A. Teredesai,"A comparative analysis of trust-enhanced recommenders for controversial Items," will be published in 3rd Int'l AAAI Conference on Weblogs and Social Media, San Jose, California, 2009.

[15] P. Massa and P. Avesani, "Trust-aware Recommender Systems," In Proc. of ACM Recommender Systems, Minneapolis, Minnesota, USA, 2007.