Application of Trust and Distrust in Recommender System: A Study

Parthasarathi Chakraborty
Assistant Professor
University Institute of Technology
University of Burdwan
Burdwan, India

Sunil Karforma
Reader
Department of Comp. Science
University of Burdwan
Burdwan, India

ABSTRACT
Recommender systems help customers to choose right product or service from large number of alternatives available on Internet. In recent time, trust becomes an important issue in designing effective recommender systems. In this paper we have studied the role of trust and distrust in designing recommender systems.

General Terms
E-Commerce, Information Retrieval, Web Mining.

Keywords
Social Trust, Distrust, Trust Inference Algorithms, Web of Trust, Recommender System.

1. INTRODUCTION
In the recent years, with the huge popularity of Web based Social Networks, the trust and trust related issues become more and more important. In a social network it is not possible for anybody to know personally all others in the network. When considering the opinion of those unknown persons it becomes necessary to device a way to know how trustworthy they are. From figure 1, it can be seen that X knows Y and Y knows Z in the network. X trusts Y by t_{XY} and Y trusts Z by t_{YZ}. Now X can use these trust information to infer how much he or she may trust Z.

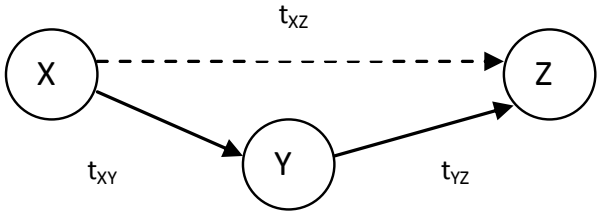

Fig 1. Example of trust inference.

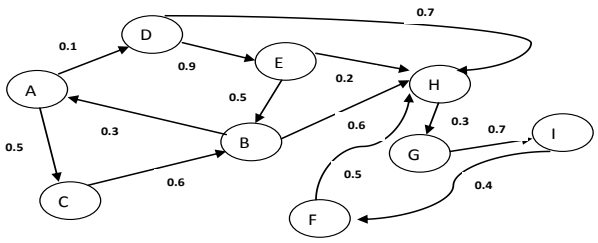

Figure 2 : Web of Trust

A trust network shows the trust relationships among the entities of the network. It can be represented by a directed graph. Nodes of the graph represent individual entities and weight on each directed edge shows how much the entity represented by source node trusts the entity represented by destination node. Figure 2 shows a trust network with nine entities.

In today's world, for providing effective recommendation to the customer, incorporating trust information into the recommender system becomes necessary. A trust based recommender system generates recommendation based on the opinion of the customer's trusted neighbors. A different strategy is to consider the opinion of all the neighbors but the degree of importance of the opinion is decided according to the trustworthiness of the neighbor to the customer. In any way, trust information, if available, helps in designing more powerful recommender systems.

Trust in web based social network, sometimes called social trust has some interesting properties. [1] propose three main properties of trust- transitivity, asymmetry, and personalization. They propose that transitivity of trust has some difference with its mathematical meaning in the sense that if Alice trusts Bob, and Bob trusts Charles, then it is not necessarily true for Alice to trust Charles in the same level of trust. Trust asymmetry indicates that trustworthiness of user1 to user2 is not same as trustworthiness of user2 to user1. By trust personalization they suggest, inherently trust is a personal opinion. Different people may have different opinion about a person. So one person may be trustworthy from one's point of view, may be completely untrustworthy from other's point of view.

2. CLASSIFICATION OF TRUST METRICS
As have been mentioned above, a trust network can be represented as a trust graph. Depending on the fact that whether the entire trust graph will be considered in calculating the trust value, the trust metrics can be divided into two categories-local trust metrics and global trust metrics.

In case of global trust metrics, a single trust value is calculated for every entity in the trust graph irrespective of the source entity for which the trust is calculated. Global trust metrics are more appropriate for situations where, for a node, a single trust value prevails throughout the system. Peer-to-peer content sharing system is one such example where trustworthiness of a node is given by a single trust value which remains same for all other nodes in the system. Table 1 shows some global trust metrics proposed by different researchers.

Table 1: Categorization of Trust metrics

Type of trust metric	Name
Global	NICE [2], Global Trust Model [3]
Local	Tidaltrust [4], Fuzzytrust [5], Moletrust [6], Levien's Advogato trust metric [7]

On the other hand, in case of local trust metrics trust score of a node (we may call sink node) varies depending on the source node for which it is calculated. This is because for calculating the trust value of a node not the entire trust graph is considered; rather a "personalized" web of trust is generated. A "personalized" web of trust shows all the direct and indirect trust relationships for the source node.

3. TRUST INFERENCE ALGORITHMS

The job of a recommender system is to provide personalized recommendations to the customers. When trust is incorporated, generating recommendation for a person, say Mr. A, needs the information that how much he trusts his neighbors. As trust is subjective in nature, person A's trust on his neighbor, say Mr. B may differ from trust of any other person on Mr. B. This is why local trust metrics are mere appropriate for designing recommender systems. Two well known local trust metrics named Mole Trust [6] and Tidal trust [4] are discussed bellow.

3.1 Mole Trust

Designing an appropriate trust metric consist of basically two different phases. The first phase dictates how the trust values will be propagated along the trust graph. The second phase deals with aggregating the propagated trust values to infer a trust score between the source and sink node. Mole Trust [6] operates in two phases. The first phase removes the cycles in the trust network to convert it into a directed acyclic graph. In the second phase, through graph walking, trust values are computed along the paths from source node to destination node. In this algorithm, trust is propagated from source node to destination node. Mole Trust algorithm is tunable through a parameter, *trust propagation horizon* to set maximum allowable length of the paths from source to destination node. The pseudo code of phase one and phase two of Mole Trust algorithm is shown bellow.

Build_modified_trust_network(source,trust network, trust_propagation_horizon)

1. start
2. distance=0
3. list_of_users[distance]=source_user
4. modified_trust_network=Φ
5. add source_user to Modified_trust_network
6. while distance<= trust_propagation_horizon then
7. do
8. Distance=distance+1;
9. List_of_users[distance]=users directly connected to List_of_users[distance-1] and not yet visited
10. For each user u in List_of_users[distance]
11. Do
12. Add node u to Modified_trust_network
13. Add all edges from List_of_users[distance-1] to node u in Modified_trust_network
14. Done
15. Done
16. End;

Calculate_trust_score_from_modified_trust_network(source_ user, modified_trust_network,trust_threshold)

1. Start
2. distance=0
3. trust(source_user)=1.0
4. While distance<= trust_propagation_horizon then
5. Do
6. distance=distance+1
7. for each user u in List_of_users[distance]
8. do
9. predecessors = list of users i having egde in u provided trust(i)> trust_threshold

$$trust(u) = \frac{\sum_{i \in predecessors} (trust\_edge(i,u) * trust(i))}{\sum_{i \in predecessors} (trust(i))}$$

Done

10. done
11. end;

Please use a 9-point Times Roman font, or other Roman font with serifs, as close as possible in appearance to Times Roman in which these guidelines have been set. The goal is to have a 9-point text, as you see here. Please use sans-serif or non-proportional fonts only for special purposes, such as distinguishing source code text. If Times Roman is not available, try the font named Computer Modern Roman. On a Macintosh, use the font named Times. Right margins should be justified, not ragged.

3.2 Tidal Trust

The Tidal Trust algorithm [4] works in the following way. If there does not exist any direct edge from source to sink, the source node asks the nodes directly connected to it to provide trust information about the sink. The source node then computes the inferred trust value as

$$t_{Source,Sink} = \frac{\sum_{i \in adjacent\_nodes(source) \wedge t_{source,i} > T} t_{source,i} t_{i,\text{sink}}}{\sum_{i \in adjacent\_nodes(source) \wedge t_{source,i} > T} t_{source,i}} \quad (1)$$

The process will continue until the path between the source node and the sink node is reached. T is a threshold value which dictates the minimum strength that should be maintained along a path. The name "Tidal" is given due to the

fact that the calculation for inferring trust is started from the sink node once the path between the source node and the sink node is found.

One significant difference between Tidal Trust and Mole Trust algorithm is, in case of Mole Trust, trust horizon can be set explicitly to specify the maximum degree of separation from the source node. Another difference is in the way of calculating the inferred trust value between the source and the sink. In Mole Trust algorithm, trust propagation follows trust aggregation where as in case of Tidal Trust, aggregation phase comes before propagation phase in a recursive manner.

4. COMPUTING TRUST FROM RATING DATA

One way to make the recommender system trustworthy is to compute trust from rating data. In [9] authors proposed the concept of profile-level and item-level trust in the context of recommender systems. In the recommendation process, for predicting the rating $\hat{r}_i$ of item i for active user, U_A, rating of several other users for that item is considered. The active user's profile is considered as consumer profile and profile of other users who take part in the recommendation process are called producer profiles. A prediction is called correct if the absolute difference of predicted rating and actual rating lies within a distance, ϵ.

If $TSet$ is the set of all predictions where the producer profile is involved and out of those predictions if $CSet$ is the set of correct predictions (as defined earlier) then profile-level trust of that customer is defined as

$$T^P(producer) = \frac{|CSet(producer)|}{|TSet(producer)|} \quad (2)$$

In reality, a producer may generate better predictions for some items over other items. Following this idea, [9] has refined equation (5.2) and defined item-level trust T^I as

$$T^I(producer, item\_i) = \frac{|CSet_i(producer)|}{|TSet_i(producer)|} \quad (3)$$

5. APPLYING TRUST IN GENERATING RECOMMENDATION

In collaborative filtering approach, after choosing the most similar k-nearest neighbors, the recommendation is made using the following formula [10] -

$$\hat{r}_{a,j} = \bar{P}_{U_a} + \left(\sum_{i\_rates\_j, i \in NU} (r_{i,j} - \bar{P}_{NU_i}) * \omega_{a,i}\right) * \left(\sum_i |\omega_{a,i}|\right)^{-1} \quad (4)$$

where $\bar{P}_{U_a}$ denotes the average ratings of active user U_A, $r_{i,j}$ is the actual rating of neighbor U_i on product I_j, $\bar{P}_{NU_i}$ denotes the average ratings of neighbor NU_i and $\omega_{a,i}$ denotes the similarity between the active user, U_A and its i^{th} neighbor, NU_i. In three ways [9] trust information can be integrated in the recommendation process.

A. Trust-based Filtering

In this approach, untrustworthy neighbors are filtered out from the list of most similar neighbors and recommendation is made only considering the ratings of the trustworthy neighbors. This scheme is named as Trust based filtering which has been shown by equation (5).

$$\hat{r}_{a,j} = \bar{P}_{U_a} + \left(\sum_{i\_rates\_j, i \in TNU} (r_{i,j} - \bar{P}_{NU_i}) * \omega_{a,i}\right) * \left(\sum_i |\omega_{a,i}|\right)^{-1} \quad (5)$$

TNU is the set of trusted neighbors and $\omega_{a,i} = sim(a,i)$ denotes the similarity between the active user, U_A and its i^{th} neighbor, NU_i.

B. Trust-based Weighting

In the second approach, importance of the ratings of the neighbors is given according to their degree of trustworthiness to the active user i.e. the user for which recommendation is made. The process is shown by equation (6).

$$\hat{r}_{a,j} = \bar{P}_{U_a} + \left(\sum_{i\_rates\_j, i \in NU} (r_{i,j} - \bar{P}_{NU_i}) * \omega_{a,i}\right) * \left(\sum_i |\omega_{a,i}|\right)^{-1}$$

where $$\omega_{a,i} = \frac{2(sim(a,i)trust(a,i))}{sim(a,i) + trust(a,i)} \quad (6)$$

where $sim(a,i)$ and $trust(a,i)$ are the similarity and the trust between the active user, U_A and its i^{th} neighbor, NU_i respectively.

C. Combining Trust-based Weighting and Filtering

This approach is the combination of the previous two approaches where only the trustworthy neighbors are considered in the recommendation process and their ratings are given weight according to their trustworthiness (shown in equation 7).

$$\hat{r}_{a,j} = \bar{P}_{U_a} + \left(\sum_{i\_rates\_j, i \in TNU} (r_{i,j} - \bar{P}_{NU_i}) * \omega_{a,i}\right) * \left(\sum_i |\omega_{a,i}|\right)^{-1}$$

where $$\omega_{a,i} = \frac{2(sim(a,i)trust(a,i))}{sim(a,i) + trust(a,i)} \quad (7)$$

Golbeck [1] proposed a different approach to calculate prediction using web of trust. If no rater of the item concerned directly connected to the source node (for which recommendation will be generated) is found then raters at the next level is searched and this process will continue until a path is found. Then trustworthiness of all the raters at the given depth is inferred using Tidal Trust algorithm and the raters with maximum inferred trust values are chosen and finally, predicted rating is calculated as average of the selected raters ratings weighted by the inferred trust values.

$$\hat{r}_{a,j} = \frac{\sum_{i \in TNU} r_{ij} t_{ai}}{\sum_{i \in TNU} t_{ai}} \quad (8)$$

6. MODELING DISTRUST

A. Bilattice Model

Along with trust information, distrust information can also play an important role in designing an effective recommender system. In presence of distrust, each trust relationship is characterized by $< t_i, d_i >$ pair where t_i represents the trust value and d_i represents the distrust value of that relationship. In presence of distrust, a Bilattice model has been proposed in [11].

A bilattice trust model [11] can be represented as a quadruple shown in equation (9).

$$BL = ([0,1]^2, \leq t, \leq k, \neg) \quad (9)$$

where $\leq t$ is the trust ordering, $\leq k$ is the knowledge ordering and $\neg$ is a $\leq t$ -negation on $[0,1]^2$.

In the bilattice trust model, the Trust Lattice, $([0,1]^2, \leq t)$ represents the trust scores (both trust and distrust information) ranging from complete distrust (0,1) to complete trust(1,0) and the knowledge lattice, $([0,1]^2, \leq k)$ represents the amount of trust evidence available between the two nodes ranging from "shortage of evidence" to "excess of evidence". If trust scores of two users are (t_1, d_1) and (t_2, d_2) respectively then shortage of evidence is represented by the fact $t_1 + d_1 < 1$ and excess of evidence is represented by the fact $t_1 + d_1 > 1$.

B. Trust Score Propagation and Aggregation Operators

As have been mentioned in section III, for estimating trust in an unknown user the trust scores of the intermediate users are considered from the source user to the target user. For this reason, proper propagation mechanism should be devised. Authors of paper [12] described a formal framework of trust-distrust propagation in a computational way. In presence of distrust, four propagation strategies have been proposed by Victor [13]. For the purpose of aggregation, they have proposed Trust Score Weighted Average Aggregation operator(T-OWA) which is based on Weighted Average Aggregation operator(OWA) [14]. GUHA [12] proposed three models of distrust propagation namely Trust Only model, One-Step Distrust model and Propagated Distrust model. In Trust Only model, distrust information is ignored totally and only the trust information is propagated. In One-Step Distrust model it is assumed that distrust is propagated only one step while trust may propagate repeatedly. The logic behind this model is if one distrusts someone else, he does not give importance to the judgment of that person. In Propagated Distrust model both trust and distrust propagates and belief, B is calculated as difference of the trust and distrust values.

7. INCORPORATING DISTRUST IN RECOMMENDATION SYSTEM

Do not include headers, footers or page numbers in your submission. These will be added when the publications are assembled.

Different ways have been proposed by the researcher in which distrust information may be incorporated in the collaborative filtering process.

A. Distrust for Filtering Neighborhood of Active User

Authors of paper [11] have argued in favor of excluding the distrusted neighbors from the recommendation process.

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u \in R^+ \setminus R^D} w_{a,u} (r_{u,i} - \bar{r}_u)}{\sum_{u \in R^+ \setminus R^D} w_{a,u}} \quad (9)$$

B. Distrust for Validating Propagated Trust Value

In this approach, the propagated trust value which is calculated from the web of trust using some trust metric is validated against the available distrust information. If it is seen that the propagated trust information contradicts the available distrust information then that propagated trust value is discarded and not taken into consideration for recommendation generation.

$$PT_{a,c} = Trust\_Metric(WOT)$$

$$Effective\_Trust(a,c) = PT_{a,c} \text{ if } PT_{a,c} \text{ does not contradict } D_{a,c} \in WOD \text{ or} \quad (10)$$

$$= 0 \text{ otherwise.}$$

C. Distrust as Negative Weight

In this approach, proposed by [11], distrusted neighbors are not excluded from the recommendation process, rather their distrust score to the active user is considered as negative weight.

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u \in R^T} t_{a,u} (r_{u,i} - \bar{r}_u)}{\sum_{u \in R^T} t_{a,u}} - \frac{\sum_{u \in R^D} d_{a,v} (r_{v,i} - \bar{r}_v)}{\sum_{v \in R^D} d_{a,v}} \quad (11)$$

D. Trust Score Based Weighted Mean (TSBWM)

Golbeck's formula (8) for generating recommendation in presence of trust has been modified by Patricia [15] in order to highlight the distrust angle also. The modified formula is given by equation (12) where all the neighbors who have rated the target product are considered in the process of recommendation generation but their importance is determined by the $(\hat{t}_{a,u} - \hat{d}_{a,u})$ value. If the distrust score exceeds the trust score for the active user, then only the neighbor is excluded from the recommendation process by setting his weight as zero.

$$p_{a,i} = \frac{\sum_{u \in R} \max(0, \hat{t}_{a,u} - \hat{d}_{a,u}) \cdot r_{u,i}}{\sum_{u \in R} \max(0, \hat{t}_{a,u} - \hat{d}_{a,u})} \quad (12)$$

8. EXPERIMENTS

The experiments have been done using movieLens dataset. We have computed the trust values using the concept of profile-level and item-level trust [9]. The ratings are arranged according to their timestamp values and a percentage of ratings are used to generate the initial profile-level and item-level trust values. Prediction are generated thereafter and compared with the actual ratings to calculate the MAE values. Based on the correctness of the predictions, the trust values are updated accordingly. The first column indicates the training/test ratio, for example 40/60 implies that the first 40% ratings are used initially to calculate the trust values. The MAE values of the 2nd column corresponds to the cases where only trust values are considered as weights in calculating predictions whereas the last column shows MAE values where only similarity values are used as weights in prediction calculation. The 3rd, 4th, 5th and 6th column shows MAE values where 80%, 60%, 30% and 10% weights have been given respectively to trust values in generating prediction.

Trust : Similarity / **Training/Test Ratio**	**Only Trust**	**80/20**	**60/40**	**30/70**	**10/90**	**Without Trust**
40/60	0.7987	0.7987	0.7989	0.8007	0.816	0.82129
60/40	0.79341	0.79359	0.7939	0.79522	0.81086	0.81716
80/20	0.79294	0.79335	0.79399	0.79614	0.82734	0.82555

Table 2 : MAE values of Trust based Collaborative Filtering for different weights of trust values and for different training/test ratios

In calculating the profile-level and item-level trust [9], the ϵ is assumed to 0.5 and number of neighbors is set to 10. The results clearly indicate that the application of profile-level and item-level trust [9] in collaborative filtering made an improvement in performance over the system without trust. There is also an indication that more and more ratings are used in calculating the initial trust values the accuracy of the recommender system is also improved.

9. CONCLUSION

In this paper, we have studied the concept of trust and distrust in the context of recommender system. We have also studied the ways of computing trust and distrust and their application in generating recommendations. In future, we are interested in detailed study on distrust propagation and it's application in designing recommender systems.

10. REFERENCES

[1] Golbeck, J, Parsia, B., Hendler, J., "Trust Networks on the Semantic Web," Proceedings of Cooperative Intelligent Agents 2003, August 27-29, Helsinki, Finland, 2003.

[2] Suryanarayana, G., H. Diallo, M., Erenkrantz,J. R. and Taylor, R. N. Architectural Support for Trust Models in Decentralized Applications. In CSE'06, May, 2006, Shanghai, China, 2006.

[3] Aberer, K. and Despotovic, Z., Managing Trust in a Peer-2-Peer Information System. In CIKM'01, November S-10, 2001, Atlanta, Georgia, USA, 2001.

[4] Golbeck, J., Hendler, J., Inferring Binary Trust Relationships in Web-Based Social Networks. ACM Transactions on Internet Technology, Volume 6, Issue 4, New York, NY,USA, 2006.

[5] Lesani, M. and Bagheri, S. Fuzzy Trust nference in Trust Graphs and its Application in Semantic Web Social Networks. World Automation Congress, 2006. WAC '06. Sharif University of Technology, Iran, 2006.

[6] Avesani, P. Massa, P. and Tiella, R., Moleskiing.it: a trust-aware recommender system for ski mountaineering. International Journal for Infonomics, 2005.

[7] Levien and Aiken. Advogato's trust metric. online at http://advogato.org/trust-metric.html, 2002.

[8] Massa, P. and Avesani, P., Trust-Aware Collaborative Filtering for Recommender Systems, Lecture Notes in Computer Science, Vol. 3290, pp. 492-508, 2004.

[9] O'Donovan, J., and Smyth, B., Trust in recommender systems. In Proceedings of the 10th International Conference on Intelligent User Interfaces, pages 167–174. ACM Press, 2005.

[10] Resinck., P., Neophytos, I., Mitesh, S., Peter, B., John, R., GroupLens: An Open Architecture for Collaborative Filtering of Netnews. Proceedings of the 1994 ACM conference on Computer Supported Cooperative Work, Chapel Hill, North Carolina, United States, p.175-186,1994.

[11] Victor, P., Cornelis, C., DE Cock, M., and Teredesai, A., Trust- and distrust-based recommendations for controversial reviews. IEEE Intell. Syst. *26*, 1, 48–55, 2011.

[12] Guha, R., Kumar, R., Raghavan, P., Tomkins, A., Propagation of trust and distrust, in: Proc. WWW2004, 2004, pp. 403–412, 2004.

[13] Victor, P., Cornelis, C., De Cock, M., Da Silva, P. P., Gradual Trust and Distrust in Recommender Systems. Fuzzy Sets and Systems 160(10), p. 1367-1382, 2009.

[14] Yager, R. R., On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decision making (1988). IEEE Transactions on Systems, Man, and Cybernetics, 18, p. 183-190, 1998.

[15] Victor. P. and Verbiest. N., Enhancing the Trust-Based Recommendation Process with Explicit Distrust, ACM Transactions on the Web, Vol. 7, No. 2, Article 6, Publication date: May 2013.