Trust based Personalized Recommender System

Punam Bedi[1]
Harmeet Kaur[2]

[1]Department of Computer Science,
University of Delhi, Delhi – 110007, India
pbedi@cs.du.ac.in

[2]Department of Computer Science, Hans Raj College,
University of Delhi, Delhi – 110007, India
harmeetnegi@hansrajcollege.com

Abstract. We rely on the information from our trustworthy acquaintances to help us take even trivial decisions in our lives. Recommender Systems use the opinions of members of a community to help individuals in that community identify the information most likely to be interesting to them or relevant to their needs. These systems use the similarity between the user and recommenders or between the items to form recommendation list for the user. They do not take into consideration the social trust network between the entities in the society to ensure that the user can trust the recommendations received from the system. The paper proposes a model where a trust network exists between the peer agents and the personalized recommendations are generated on the basis of these trust relationships. The recommenders personalize recommendations by suggesting only those movies to user that matches its taste. Also, the social recommendation process is inherently fuzzy and uncertain. In the society, the information spreads through word-of-mouth and it is not possible to fully trust this information. There is uncertainty in the validity of such information. Again, when a product is recommended, it is suggested with linguistic quantifiers such as very good, more or less good, ordinary, and so on. Thus, uncertainty and fuzziness is inherent in the recommendation process. We have used Intuitionistic Fuzzy Sets to model such uncertainty and fuzziness in the recommendation process.

Keywords. Degree of trust, Intuitionistic Fuzzy Sets, Unintentional encounters, Intentional encounters.

(Received November 18, 2005 / Accepted February 15, 2006)

1. INTRODUCTION

In our daily life, even to decide upon simple things like which movie to watch, which book to read, which restaurant to eat at, we depend upon our acquaintances, reviews in the newspapers, magazines, and general surveys, etc to help us find what is good for us. This support from the society provides us a shortcut to search out a good option without putting much effort in sifting through the various options available in the market. In this age of technology, the Recommender Systems (RS) have come to the rescue of the users that create a technological proxy for this by drawing on user preferences and filtering the set of possible options to a more manageable subset [10, 12].

The existing recommender systems do not base their recommendations on the trust relationships that exist in the society, rather suggest products on the basis of similarity between the users or the items [7]. They ignore the social elements of decision-making and advice seeking, and hence the system model does not match the mental model of the user [5]. The user agent does not know about the people whose tastes are used to suggest products that may be of interest to the user, and this result in lack of trust on the recommendations received from the system. It is found in [11] that given a choice between recommendations from friends and recommender systems, in terms of quality and usefulness, friends' recommendations are preferred even though the recommendations given by the recommender system have high novelty factor. Friends are seen as more qualified to make good and useful recommendations as compared to recommender systems.

In our recommender system, a social structure exists between the agents in the application domain, which is formed on the basis of trust them. The agents recommend movies to each other using this social structure. This is similar to the mental model of decision making of a human. The concept of trust in the recommender system has been incorporated in [8, 9], but it suffers from the problem of being highly computation intensive as it not only computes the similarity between the user agent and the peers but also computes trust values between them. Also the system requires that the trust value to be manually entered and maintained. The system on its own does change trust values set by the user, however in our model after manual initialization, the agents learn about the trustworthiness of other agents by interacting with them and appropriately modify the trust values for future interactions without the need of human intervention.

In real life we come to know about others through our social circle. However, it is not possible to decide as to what extent the piece of information that is obtained via third party is correct and as a result there is uncertainty in the recommendation if it is based on third party version of information. In addition to this, the recommendation process is inherently fuzzy. The recommendations about the products are given using the linguistic quantifiers such as very good, good, more or less good, ordinary, etc. In literature not much work is done regarding the utilization of fuzziness and the uncertainty in the recommendation process, even though these are inherent in it. The Intuitionistic Fuzzy Sets (IFS) [1] having degree of membership, degree of non-membership and degree of uncertainty are very well suited for modeling fuzziness and uncertainty in the recommender systems.

In the recommender system that we have proposed, the recommenders suggest the list of movies filtered on the basis of the tastes of the user. This filtering of the movies according to the tastes of the user *personalizes* the recommender system. The user agent then aggregates these lists to form a single list and then decides whether the movies in the list are worth watching or not. During the aggregation process, the user agent takes into consideration the IFSs for the movies which are in the form of degrees of membership, non-membership and uncertainty provided as a recommendation by the recommender. The degree of trust [4] on the recommender and the rank of the movies in the lists are the other factors in the aggregation process of the recommendation lists.

The organization of the paper is as follows. In section 2, our trust based personalized recommender system is discussed. Section 3 discusses a case study and finally section 4 concludes the paper.

2. Personalized Recommender System based on Trust

In this section we have proposed a recommender system to suggest movies to the user that incorporates the social recommendation process based on trust. The social recommendation process is taken into consideration by forming a network of the agents that act as a society and these agents interact with each other on the basis of trust relationships. These trustworthy relationships form a web of trust [6] (Fig. 1). An agent, A seeks recommendations from the agents connected to it directly and if the human user connected to that agent liked a movie then the agent gives the feedback to those peers who had recommended that particular movie. Based on this feedback, the peers update the list of preferences for the user agent A. It is not possible for a recommender to watch all the movies and then recommend few out of them. Rather the recommender comes to know about many movies through its set of acquaintances. Similarly, the recommender agents in order to recommend a movie further take the help from their trustworthy acquaintances in getting the information about new movies and so on. The social network hence formed helps to spread information through "word-of-mouth".

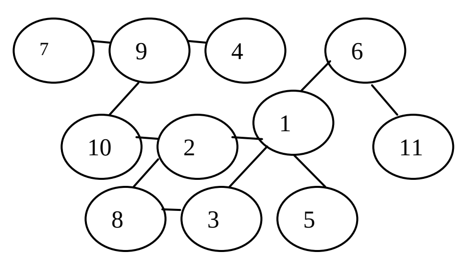

Fig. 1: Web of Trust

Fig 1 shows such a network of peers represented by the numbered circles, where the numbers in the circles identify the various peers in the application domain. An edge represents the agents that are connected directly and have a trustworthy relationship with certain degree of trust. However, it is not necessary that if A trusts B with degree of trust as *x* then B also trusts A with *x* degree.

Every agent in the system maintains a degree of trust and information about the tastes of the agents that are connected to it directly. The recommenders pass on only those recommendations to the user agent that matches its tastes leading to the *personalization* of the *recommender system*. This reduces the number of recommendations that need to be given to the user agent by removing the unnecessary recommendations and this further reduces the number of computations that the user agent has to perform at the time of aggregation of the recommendations to find something useful for itself.

The agents in this social setup exchange information about the movies and the likings of the agents known to them during their idle time, which we are referring to as the *unintentional encounters* [3]. Since the idle agents are free to join the group of agents exchanging information, such interactions are the unintentional encounters. When an agent wants to find interesting movies for itself and explicitly seeks the recommendations from its trustworthy agents, the interactions are termed as *intentional encounters*.

In our system, even though the interactions are between the agents in the system, if needed human input is also taken where it is not feasible for an agent to compute it. For example, when the feedback for movie is to be generated then it is not possible for the software agent to decide whether the movie is good or bad for the human corresponding to that agent. In such cases the human input is taken to generate the feedback. The user is the human corresponding to user agent, which is a software entity. Similarly recommender is the human corresponding to the recommender agent, again a software entity. The user is the one who is interested in finding some useful movie for himself / herself.

2.1 Generating recommendations as a recommender for user agent

The recommender agents accumulate the information during the unintentional encounters that is passed as a recommendation to the user agent during the intentional encounters. Every recommendation corresponds to a movie and is in the form of an IFS. The IFS recommendation for a movie has a degree of membership (satisfaction), degree of non-membership (dissatisfaction) and degree of hesitation (uncertainty) signifying the relevance of the movie for the user. To personalize the movie recommendations according to the tastes of the user agent A, the recommender agent maintains the following lists:

- *Preference list:* The preference list, P_A consists of the information (directors, actors, actresses and genre) about the movies liked by the user connected to A. There are separate sublists in P_A corresponding to the groups of directors, actors, actresses and genre. The order of the names in the respective groups of directors, actors, actresses and genre, signify their priority in their respective sublists.
- *Uncertain list*: This list U_A consists of the same type of information as that of the preference list, but the data about the tastes of A as accumulated by the agent during the unintentional encounters and via the feedback process. However, there is no prioritization among the groups of directors, actors, actresses, or genre as this list is accumulated during the unintentional encounters and the recommender agent has no idea whether the user prefers one actor over the other and so on.

In this paper, we are trying to have a system similar to the social recommendation process and hence we are not restricting to the preference list or uncertain list for the user tastes. As in real life, to recommend a movie to someone known to us, we do take into consideration the tastes of the person. But if we feel that a particular movie may be of interest to the other person as the movie has a general appeal, we do recommend that movie. In such cases, if the user likes the movie that actually does not conform to his/her tastes explicitly mentioned, then the user agent gives a feedback to the recommender agent(s) who recommended that particular movie. The recommender agent on getting a positive feedback from the user agent adds the name of the directors, actors, actresses and the genre of the movie to the uncertain list for that user.

Through any of the following ways a movie can become a candidate of the recommendation:

- *Case 1: the recommender has seen the movie*, and feels that the movie has general appeal even if it does not conform to the tastes of the user. The recommender explicitly instructs the recommender agent to recommend that movie to the user. The degree of uncertainty for the IFS of this movie is provided by the recommender. The degree of hesitation signifies the extent to which the recommender is not sure about his/her decision to suggest that movie to the user. The degree of membership is zero for such movies and the third parameter is computed using the other two degrees. All such movies are recommended to the user.
- *Case 2: the recommender agent comes to know about the movie through a trustworthy acquaintance during unintentional encounters*.
- *Case 3: the movie is in the database of the recommender agent.*

The movies of case 1 are recommended whether they are according to the tastes of the user or not. For the movies of case 2 and 3, matching is done with P_A and U_A and the movies that do not match any of these lists are not recommended to the user.

2.1.1 IFS Generation for the movies

The IFS for movies of case 1 has degree of membership to be zero and the degree of uncertainty is provided by the recommender. The IFS for the movies of case 2 and case 3 is computed as follows:

1. Form a single list for all the movies of case 2 and case 3, and for each movie perform step 2 to 5.
2. The degree of membership of movie M, μ_M is computed using the preference list P_A, as:
   2.1. Let there be x number of directors (d_1, d_2, …, d_x) of the movie M. Search the names of these x director(s) in the directors' sublists of P_A.
   2.2. If d_i (i = 1, 2, …,x) figures in the list then compute the rank r_{di} as the position of d_i in the directors' sublist, else r_{di} is 0.
   2.3. Similarly compute the ranks of genre and all the actors and actresses of the movie M in their respective sublists of P_A. Let the rank of genre of M be r_g. Let there be y actors of M with the ranks as r_{a1}, r_{a2}, …, and r_{ay}. Similarly, let there be z actresses of M with the ranks as r_{ac1}, r_{ac2}, …, and r_{acz}.
   2.4. Finally,
   $$\mu_M = (g * (r_g) + d * (r_{d1} + r_{d2} + \ldots + r_{dx}) + a * (r_{a1} + r_{a2} + \ldots + r_{ay}) + ac * (r_{ac1} + r_{ac2} + \ldots + r_{acz})) / (t_g + t_d + t_a + t_{ac}) \quad (1)$$
   where g, d, a and ac represent the degrees of significance that the user associates with the subgroups of genre, directors, actors and actresses, respectively, and
   t_g, t_d, t_a and t_{ac} represent the total number of genre, directors, actors and actresses that are present in the respective sublists of P_A.
3. The degree of uncertainty of movie M, π_M is computed using the uncertainty list U_A, as:
   3.1. Let there be h number of genre, i number of directors, j number of actors and k number of actresses in the uncertain list. Let p directors of M be present among the list of i directors of the uncertain list. Similarly, let q actors and r actresses of M be present in their respective lists of actors and actresses in the uncertain list.
   3.2. Compute the degree of uncertainty of the movie M as:
   $$\pi_M = (g * f + d * p + a * q + ac * r) / (h + i + j + k) \quad (2)$$
   where, f is 1 if the genre of M is in the uncertain list else it is 0, and
   g, d, a and ac is same as above.
4. The degree of non-membership of movie M, ν_M is compute as follows:
   $$\nu_M = 1 - \mu_M - \pi_M \quad (3)$$
5. The movies with degree of membership, μ_M = 0 and degree of uncertainty, π_M = 0 are not considered for further processing.

2.1.2 Final recommendation list generation

After matching the movies with the preference list and uncertain list, the degree of membership, non-membership and uncertainty is available with the recommender agent for all the movies that it knows. The following method is used to generate the final list of the movies that are to be recommended to the user agent along with IFS that is computed for them:

1. All the movies that are a part of case 1 are to be considered for further processing.
2. For all the movies of case 2 and case 3 that are to be considered, do the following:
   2.1. The movies with non-zero degree of uncertainty are followed by the movies with non-zero degree of membership.
   2.2. Within the movies with non-zero degree of uncertainty, order the movies in ascending order on degree of uncertainty.
   2.3. Within the movies with non-zero degree of membership, order the movies in ascending order on degree of membership.

2.2 Aggregation of recommendation lists after intentional encounters by the user agent

The user agent need to form an aggregated order out of the IFS recommendation lists received from various sources during the intentional encounters. The user agent has to generate a final consolidated list from all the recommendations that are received from the recommenders. The user agent computes the degree of importance of a movie on the basis of degree of trust on the recommenders who have recommended the movie, the relative position of the movie in the list of the recommenders and the IFS recommendation of the recommender. From this aggregated list the user agent can take a decision whether or not to watch the movies suggested by the recommenders. The user agent generates a final consolidated list from all the recommendations that are received from the recommenders using the following aggregation method:

1. First identify the distinct movies from the lists and then compute the degree of importance (DoI) of every movie (M_i) as follows:

$$\begin{aligned} DoI_i(A) = {} & DoT(R_1) * \{\mu_i(R_1) - \nu_i(R_1) * \pi_i(R_1)\} * Rank_i(R_1) \cap \\ & DoT(R_2) * \{\mu_i(R_2) - \nu_i(R_2) * \pi_i(R_2)\} * Rank_i(R_2) \cap \ldots \cap \\ & DoT(R_k) * \{\mu_i(R_k) - \nu_i(R_k) * \pi_i(R_k)\} * Rank_i(R_k) \end{aligned} \quad (4)$$

   where, $DoI_i(A)$ is the degree of importance of M_i as computed by A,
   $\cap$ is the fuzzy intersection operator,
   R_j is the j^{th} recommender,
   $\mu_i(X)$ is the degree of membership of M_i according to X,
   $\nu_i(X)$ is the degree of non-membership of M_i according to X,
   $\pi_i(X)$ is the degree of uncertainty or hesitation of M_i according to X,
   $DoT(R_j)$ is the degree of trust of the A on R_j,
   $Rank_i(R_j)$ is the normalized position of M_i in the recommendation list of R_j,
   k is the total number of recommenders who have recommended M_i.
2. Arrange the movies in the ascending order of their degrees of importance as obtained in equation (4).

The degree of importance is negative for those movies that do not conform to the user tastes exactly. They have been recommended as they have mass appeal or it has matched only the uncertain list and not the preference list. The user is free to select any movie from the aggregated list.

2.3 Updating Degree of Trust of the recommenders

The degree of trust on a recommender is updated on the basis of the distance between degree of importance of the movie as it is there in the aggregated list of the user agent, A and the recommendation list of the recommender, R [2]. The distance between A and R, d signifies the *degree of similarity* between the user and the recommender and is computed as follows:

$$d = (|D_1| + |D_2| + \ldots + |D_p|) / p \quad (5)$$

where, $D_i = \{\mu_i(R) - \nu_i(R) * \pi_i(R)\} - \{\mu_i(A) - \nu_i(A) * \pi_i(A)\}$, and

p is the total number of movies in the recommendation list of R.

Depending upon whether the difference between its aggregated list and the recommendations is below its acceptable threshold d_t or not, the user agent updates the degree of trust, DoT(R) on recommender as follows:

$$DoT(R) = DoT(R) + (d_t - d) \quad (6)$$

In our model, in this way the degrees of similarity between the agents get absorbed into the corresponding degrees of trust, thus making the computation of degree of similarity between the user and the recommenders redundant.

3. CASE STUDY

An experiment was conducted in which five friends were asked to help the authors decide about which movie to watch at the weekend. The author has a certain degree of trust on these friends, which is represented in the Table – 1. In Table – 2, degrees of significance that the user associates with the subgroups: genre, directors, actors and actresses are mentioned. The Table – 3 gives the preference list that the authors gave to the recommender friends. The information in the Table – 4 is what the recommenders know, considering the nature of the author. The recommenders responded with the degrees of membership and non-membership about the movies as shown in the Table – 5. Finally, in Table – 6, the aggregated list of all the recommendations as computed by the user is given. For the remaining part of the case study, the author will be referred to as the user and the friends will be referred as the recommenders.

Recommenders	1	2	3	4	5
Degree of Trust	0.89	0.64	0.85	0.73	0.93

Table 1: The degree of trust on the recommenders according to the user

Genre (g)	*Directors(d)*	*Actors(a)*	*Actresses(ac)*
0.3	0.4	0.2	0.1

Table 2: Degrees of significance of the subgroups: genre, directors, actors and actresses

Sublists	*Preferences in the sublists*				
Genre	Romantic	Comedy	---	---	---
Directors	Karan Johar	Ram Gopal Verma	Farhan Akhtar	Priyadarsha n	David Dhavan
Actors	Shah Rukh Khan	Amitabh Bachan	Sanjay Dutt	Hrithik Roshan	Ajay Devgan
Actresses	Kajol	Rani Mukerji	Priety Zinta	Aishwarya Rai	---

Table 3: Preference List (all the sublists) about tastes of user maintained by recommenders

Recommender / *Sublists*	*1*	*2*	*3*	*4*	*5*
Genre	Thriller	Thriller, Horror	Thriller	Thriller, Horror	Thriller
Directors	Rakesh Roshan	Yash Chopra	Rakesh Roshan	Rakesh Roshan	Sanjay Leela Bhansali
Actors	Saif Ali Khan	Aamir Khan	Aamir Khan	Saif Ali Khan	Salman Khan
Actresses	Bipasha Basu	Esha Deol	Amisha Patel	Amisha Patel	Bipasha Basu

Table 4: Uncertain lists about the user as maintained by the five recommenders

Recommender / *Movie Recommended*		*1*	*2*	*3*	*4*	*5*
Aks	μ	0	0			
	ν	0.45	0.37			
Bhoot	μ		0			0
	ν		0.42			0.6
Chalte Chalte	μ	0.13571		0.13571		
	ν	0.86429		0.86429		
Dil Chahta Hai	μ			0.14286		0.14286
	ν			0.807143		0.857143
Dil Wale Dulhaniya Le Jayege	μ		0.14286		0.14286	
	ν		0.85714		0.85714	
Hanuman	μ			0		
	ν			0.2		
Hera Pheri	μ		0.0079		0.0079	
	ν		0.9921		0.9921	
Hum Dil De Chuke Sanam	μ	0.05			0.05	
	ν	0.95			0.95	
Jodi No. 1	μ			0.09286	0.09286	0.09286
	ν			0.90714	0.90714	0.90714
Kal Ho Naa Ho	μ		0.11443		0.11443	
	ν		0.885572		0.8357	
Koi Mil Gaya	μ	0	0	0		
	ν	0.73	0.45	0.45		
Kuch Kuch Hota Hai	μ		0.30714	0.30714		0.30714
	ν		0.692858	0.692858		0.692858
Mangal Pandey	μ			0		0
	ν			0.6		0.2
Munna Bhai MBBS	μ	0.064		0.064		
	ν	0.9358		0.9358		
Shaadi No 1	μ				0.0929	0.0929
	ν				0.9071	0.9071

Table 5: Recommendations of the five recommenders

Movie	*Degree of Importance*
Dil Wale Dulhaniya Le Jayege	0.104082
Chalte Chalte	0.074303
Kuch Kuch Hota Hai	0.068865
Kal Ho Naa Ho	0.056178
Munna Bhai MBBS	0.023433
Hum Dil De Chuke Sanam	0.015500
Dil Chahta Hai	0.007794
Shaadi No 1	0.003561
Jodi No. 1	0.002005
Hera Pheri	0.001218
Mangal Pandey	-0.02190
Hanuman	-0.04380
Koi Mil Gaya	-0.04517
Bhoot	-0.05916
Aks	-0.08492

Table 6: The aggregated list as obtained by the user

The degree of importance is negative for those movies that do not conform to the tastes of the user but have been suggested as they have mass appeal. The user can select any movie from those recommended.

4. Conclusions

The existing recommender systems base recommendations on similarity between the user and the recommenders or between the items. In the first case, the user profile is matched with the database of profiles to find the similar profiles. The products preferred by those similar people are suggested to the user also. In the second type of recommender systems, the database is mined to find products that are normally preferred together. Depending upon what user has already purchased/shown preference for; the other products that go along with it are suggested to the user. However, the studies have shown that the users prefer recommendations from friends as compared to the recommendations received from these recommender systems. This is because the existing recommender systems work like a black box and hence it is difficult for the user to accept the recommendations of the system. To overcome this problem of lack of trust on the recommendation systems we have proposed a model that incorporates the social recommendation process. The trustworthy peers of the user become the recommender agents and suggest movies to the user according to the tastes of the user. The agents in our system also learn from their experience in dealing with the trustworthy peers and update the degree of trust on them. In the proposed system, we have tried to merge the advantages of the mechanical recommender system with the more humane recommendation process to make their recommendations trustworthy and useful for the user.

References

1. Atanassov K.. *Intuitionistic Fuzzy Sets: Theory and Applications*, *Studies in Fuzziness and Soft Computing*. Vol. 35, Physica-Verlag , 1999.
2. Bedi P. and Kaur H. *Using Fuzzy Clustering to Determine Trust based Recommendations*. Accepted for publication in the *Proceedings of Indian International Conference on Artificial Intelligence*, Dec. 2005, Pune, India, p. 2120 – 2136, 2005.
3. Bedi P. and Kaur H. *Trust Based Recommender System*. In the *Proceedings of International Conference on Artificial Intelligence*, Las Vegas, USA. p. 798 – 801, 2005.
4. Bedi P. and Kaur H. *Fuzzy Quantification of Trust*. In the *Proceedings of International Conference on Cognitive Systems*. New Delhi, India, 2004
5. Bonhard P. *Who do trust? Combining Recommender Systems and Social Networking for Better Advice*. In the *Proceedings of the Workshop Program at International Conference on Intelligent User Interfaces*. San Diego, California, USA, 2005.
6. Guha R., Kumar R., Raghavan P. and Tomkins A. *Propagation of Trust and Distrust*. In the *Proceedings of World Wide Web*, New York, USA. p. 403 – 412, 2004.
7. Karypis G. *Evaluation of Item-Based Top-N Recommendation Algorithms*. In the *Proceedings of the tenth International Conference on Information and Knowledge Management*, ACM Press, New York, USA, 2001.
8. Massa P. and Avesani P. *Trust-aware Collaborative Filtering for Recommender Systems*. In the *Proceedings of International Conference on Cooperative Information Systems*, 2004.
9. Massa P. and Bhattacharjee B. *Using Trust in Recommender Systems: an Experimental Analysis*. In the *Proceedings of iTrust*, Oxford, UK, Springer, Vol 2995, p. 221 – 235, 2004.
10. Resnick P. and Varian H.R. *Recommender Systems, Communications of the ACM*. Vol. 40(3), p. 56 – 58, 1997.
11. Sinha R. and Swearingen K. *Comparing Recommendation made by Online Systems and Friends*. In the *Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries*, Ireland, 2001.
12. Swearingen K., and Sinha R. *Interaction Design for Recommender System*. In the *Proceedings of Designing Interactive Systems*, London, 2002.