ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/220815866

Trust Based Recommender System for Semantic Web.

Conference Paper · January 2007

Source: DBLP

CITATIONS
123

READS
98

3 authors:

Punam Bedi
University of Delhi
217 PUBLICATIONS **1,347** CITATIONS
SEE PROFILE

Harmeet Kaur
University of Delhi
41 PUBLICATIONS **236** CITATIONS
SEE PROFILE

Sudeep Marwaha
Indian Agricultural Statistics Research Institute
22 PUBLICATIONS **155** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Building and Querying Ontology for Agriculturally Important Microbes View project

Recommender system View project

All content following this page was uploaded by Harmeet Kaur on 19 May 2014.

The user has requested enhancement of the downloaded file.

Trust based Recommender System for the Semantic Web

Punam Bedi, Harmeet Kaur, Sudeep Marwaha
Department of Computer Science, University of Delhi, Delhi -110007, India
pbedi@cs.du.ac.in, hkaur@cs.du.ac.in, sudeep@iasri.res.in

Abstract

This paper proposes the design of a recommender system that uses knowledge stored in the form of ontologies. The interactions amongst the peer agents for generating recommendations are based on the trust network that exists between them. Recommendations about a product given by peer agents are in the form of Intuitionistic Fuzzy Sets specified using degree of membership, non membership and uncertainty. In literature, the recommender systems use databases to generate recommendations. The presented design uses ontologies, a knowledge representation technique for creating annotated content for Semantic Web. Seeing the potential and popularity of ontologies among researchers, we believe that ontologies will be build and maintained in numerous knowledge domains for the Semantic Web and future applications. The presented recommender system uses temporal ontologies that absorb the effect of changes in the ontologies due to the dynamic nature of domains, in addition to the benefits of ontologies. A case study of tourism recommender system is chosen to generate the recommendations for the selection of destination, travel agents and the flight schedule. A comparison of the generated recommendations with the manual recommendations by peers establishes the validity of the presented recommender system.

1 Introduction

In our daily life, even to decide upon simple things like which place to visit, 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 find interesting products or services. This support from the society provides a shortcut to select a good alternative as otherwise the cost and effort required is usually not deemed to be worth the trouble.

In this age of technology, the Recommender Systems (RS) have come to the rescue of the users and create a technological proxy for this. Recommender Systems use the opinion of members of a community to help individuals identify the information most likely to be interesting to them or relevant to their needs. This is done by drawing on user preferences and filtering the set of possible options to a more manageable subset. Collaborative filtering is the most common technique used by the recommender systems, in which the products are suggested to the user on the basis of users or items similarity [Herlocker et al., 2000; Karypis, 2001]. However, such recommender systems ignore the social elements of decision-making and advice seeking, and hence the system model does not match the mental model of the user [Bonhard et al., 2006]. The user does not know about the taste of the people that form the basis to suggest products. This acts as a hindrance to trust the recommendations of the system.

[Sinha et al., 2001] have shown 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 systems have high novelty factor. Friends are seen as more qualified to make good and useful recommendations as compared to recommender systems. Also, the empirical studies by [Ziegler et al., 2004] have shown that the correlation exists between trust and user similarity when the community's trust network is bound to some specific application. In light of these studies, it can be said that the computational trust models can act as appropriate means to supplement [Donovan et al., 2005] or replace [Ziegler et al., 2004] current collaborative filtering approaches used by the recommender systems.

To overcome the limitations of existing recommender systems, we have proposed a trust based recommender system for the Semantic Web. The proposed recommender system runs on the server with the knowledge distributed over the network in the form of ontologies. An individual interacts with the recommender system through an agent. The agents of the application domain form a "web of trust" [Guha et al., 2004] and use this web of trust to generate the recommendations. In the proposed model, the recommendations are taken from the trustworthy agents only and the data as well as the methods used to generate the recommendations are with agents, making the recommendation process transparent to the user.

[Massa et al., 2004; Ziegler et al., 2004] have proposed recommender systems that use trust to recommend products. In their work, trust values are computed in addition to similarity measures between the agents. In our work, similarity between the agents gets absorbed into trust through the process of trust update.

The presented design uses ontologies, a knowledge representation technique for creating annotated content for Semantic Web. Seeing the potential and popularity of ontologies among researchers, we believe that ontologies will be built and maintained in numerous knowledge domains for Semantic Web and future applications. The proposed architecture also enables the same agent to give recommendations on more than one subject domain. The knowledge being put as ontology on a separate tier, allows the flexibility of using more than one ontology to give recommendations on different domains or to use more than one domain to generate recommendations on a single complex problem. The presented recommender system uses temporal ontologies that absorb the effect of changes in the ontologies due to the dynamic nature of the domains.

In the literature not much work has been done regarding the utilization of fuzziness and uncertainty in the recommendation process, even though these are inherent in it. In the proposed model we have used Intuitionistic Fuzzy Sets (IFS) [Atanassov, 1999] that have degree of membership, non-membership and uncertainty to capture the fuzziness and uncertainty.

The organization of the paper is as follows. In section 2, knowledge representation using temporal ontologies is discussed. Section 3 presents our trust based recommender system for the Semantic Web. In section 4, formation of "web of trust" is described. A case study of tourism recommender system is presented in section 5 and finally section 6 concludes the paper.

2 Knowledge Representation in Temporal Ontologies

Ontology is a conceptualization of a domain into a human understandable, but machine readable format consisting of entities, attributes, relationships and axioms [Middleton et al., 2002]. The degree of specification of the conceptualization, which underlies the language used by a particular knowledge base, varies in dependence of the purposes. An ontological commitment is thus a partial semantic account of the intended conceptualization.

The OWL, Web Ontology Language is a language for defining and instantiating Web Ontologies. In OWL ontology, concepts are arranged in hierarchical format with each concept represented by a node in the hierarchy. An OWL class having various properties and relationships with the other classes represents each node. The temporal ontologies are implemented in OWL ontologies with the introduction of new attributes to classes and properties for marking timestamp and their validity.

Relating it with the frames and slots, a class in ontology is based on the frame and its properties are slots of the frames. The relationships among different classes or frames are established by referencing related classes or instances of classes in slots or properties. The temporal ontologies are obtained through frame and slot versioning [Bedi et al., 1993; Noy et al., 2004]. When the value of a property of a class has changed or name of the property has changed between two versions, we use the slot versioning to capture the change. In slot versioning, only the version of changed property is created and inserted above the existing latest version in the same OWL file. When the class name or the intrinsic attribute of the class has changed then we use the frame versioning and the whole is inserted above the existing version in the same OWL file.

3 Trust based Recommender System for Semantic Web

The framework of the recommender system shown in Fig. 1 uses temporal ontologies. It depicts the relationships between the agents and the ontologies. There can be m agents in the application domain that interact with each other forming a social network of agents based on trust, referred to as "web of trust". Every agent has its personal temporal ontology which is populated using one or more of the n domain temporal ontologies. It is not necessary that every agent

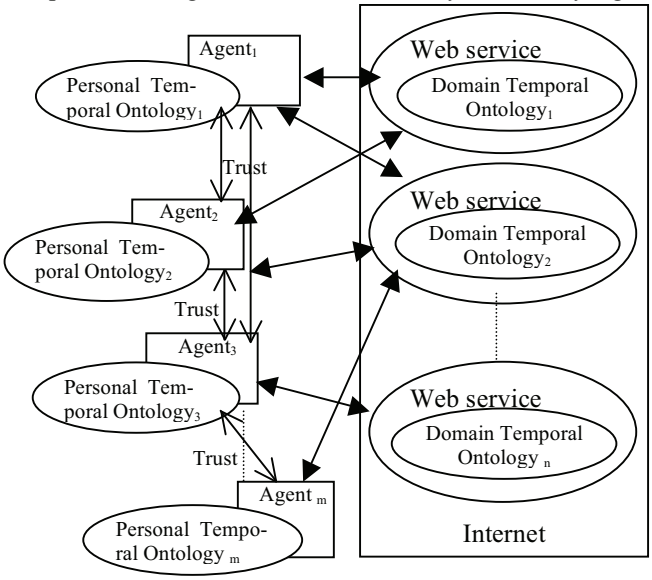

Fig.1. Trust based Recommender System for Semantic Web

interact with every agent or domain temporal ontology available in the application domain.

The domain specific temporal ontology provides the part of a specific version of the ontology or requisite version of ontology or a full temporal ontology as per the need of the requesting agent. Extraction of the sub or full ontology from the temporal ontology is taken care by the web service that is also used as an intermediate layer for handling the interactions between the agent community over the web and domain temporal ontology. In this paper, we are limiting our self to the OWL temporal ontology but the framework poses no restriction for the use of temporal ontologies written in other languages. The web service layer also provides an abstraction layer that isolates the domain temporal ontologies from the agent community. This abstraction allows the ontology engineers to create, develop, and update the ontology separately and then seamlessly integrate them with the agent applications [Bedi et al., 2005]. The use of temporal

ontologies in this respect is very useful as it allows the agents in recommender system to extract knowledge from different versions of the same ontology dynamically, taking care of the changes occurring in them. The agents can continue to work intelligently by using older compatible version and yet use new version for the concepts that are added afterwards but compatible with the existing ones. This architecture also provides a great scalability to scale up recommender system developed for one domain to many domains.

Using the personal ontology, a recommender agent generates personalized recommendations for the user agent. In the personal ontology, an agent maintains the profile of the acquaintances, requisite part retrieved from the domain temporal ontology and degree of trust on the acquaintances which is computed and updated as shown in section 4.

Two types of interactions / encounters are possible between the agents: *intentional and unintentional* [Bedi et al., 2006]. When an agent intends to find interesting products for itself and explicitly seeks recommendations, the interactions are termed as intentional encounters. The agents in the application domain through "web of trust" exchange information about the products known to them during their idle time. Such interactions are referred to as the unintentional encounters. The unintentional encounters help to spread information similar to "word of mouth".

An agent can act both as a recommender as well as user agent. During the intentional encounters, one agent acts as a user agent and those known to it act as recommenders. The following sections describe how the recommendations are generated as recommender and how the user agent aggregates the recommendations to generate the list of interesting products for itself.

3.1 Generating recommendations

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

- Preference list: The preference list, P_A consists of the information in terms of the attributes of the products liked by the human user connected to A. For example, in case of movie recommender system, the attributes can be directors, actors, actresses, etc. There are separate sub-lists in P_A corresponding to every attribute of the product. The order of the values in the respective attribute list, signify their priority in the 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 taste of A is acquired via the feedback process mentioned below. However, there is no prioritization among the values of an attribute as recommender agent has no idea about the user preference of one value over the other.

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 taste. As in real life, to recommend a product to someone known to us, we do take into consideration the taste of the person. But if we feel that a particular product may be of interest to the other person as the product has a general appeal or if the reviews for a product in the newspaper, magazines, etc. are good, then we do recommend that product. In such cases, if the user likes the product that actually does not conform to his/her explicitly mentioned taste, then the user agent gives a feedback as a binary value, yes or no, to the recommender agent(s) who recommended that particular product. The recommender agent on getting a positive feedback from the user agent adds the attribute values of the product to the uncertain list for that user. The user agent does not rate the product in the feedback; as a result it is not possible for the recommender agent to adjust it in the preference list. This also helps to overcome the cold start problem generally faced by the recommender systems.

Generating IFS for the products

A recommender can recommend products known to it. The recommender agent comes to know about the product either through usage or through unintentional encounters. During the unintentional encounters, an agent exchanges the information about only those products that it has used and is satisfied with.

An agent stores only the names of the products known to it in the personal domain ontology. When an agent has to generate recommendations for other agent, it retrieves knowledge about known products from the appropriate version of the domain temporal ontology.

Let the products be represented by n attributes (a_1, a_2, …, a_n). A product P is suggested to the user agent A, along with the IFS generated for it as shown below:

1. The degree of membership of product P, μ_P is computed using the preference list P_A, as:
   - 1.1 For every attribute a_i (i = 1, …, n) do the following:
     - 1.1.1 Let av_{i1}, av_{i2}, …, $av_{i(mi)}$ be the attribute values of P for the attribute a_i. Search for these values in the sublist of attribute a_i of P_A.
     - 1.1.2 If av_{ij} (j = 1, 2, …, mi) figures in the list then compute the rank rav_{ij} as the position of av_{ij} in the a_i sublist, else rav_{ij} is 0.
     - 1.1.3 Finally,

$$\mu_P = \frac{(da_1 * (rav_{11} + rav_{12} + \ldots + rav_{1(m1)}) + da_2 * (rav_{21} + rav_{22} + \ldots + rav_{2(m2)}) + \ldots + da_n * (rav_{n1} + rav_{n2} + \ldots + rav_{n(mn)}))}{t_t + t_2 + \ldots + t_n} \quad (1)$$

where da_i (i = 1, 2, …, n) represents the degree of significance that the user associates with the i^{th} attribute,
t_i (i = 1, 2, …, n) represents the total number of values that are present in the i^{th} attribute's sublists of P_A.

2. The degree of uncertainty of product P, π_P is computed using the uncertainty list U_A, as:
   2.1 Let there be u_i (i = 1, 2, …, n) entries in the i^{th} attribute's sublists in U_A.
   2.2 Let k_i be the attribute values of P for attribute a_i that are present among u_i entries.
   2.3 Compute the degree of uncertainty of the product P as:

$$\pi_p = \frac{(da_1 * k_1 + da_2 * k_2 + ... + da_n * k_n)}{u_1 + u_2 + ... + u_n} \quad (2)$$

   where, da_i (i = 1, 2, …, n) is same as in step 1.1.3.
3. The degree of non-membership of product P, ν_P is compute as follows:

$$\nu_p = 1 - \mu_p - \pi_p \quad (3)$$

Final recommendation list generation

After matching the products with the preference list and uncertain list, the degree of membership, non-membership and uncertainty is available with the recommender agent for all the products that it knows. If a product, P has both μ_P and π_P as zero, then the product is not considered further unless and until the recommender feels that the product has general appeal or the reviews for it are good. The recommender provides the degree of uncertainty for the IFS of such products that signify the extent to which the recommender is not sure about his/her decision to suggest that product to the user. The degree of membership is zero for such products and the third parameter is computed using eq. (3). The following method is used to generate the final list of the products that are to be recommended to the user agent along with IFS that is computed for them:

1. The products having membership as zero and uncertainty as non-zero are followed by the products with non-zero degree of membership.
2. Within the products with membership as zero and uncertainty as non-zero, order the products in ascending order on degree of uncertainty.
3. Within the products with non-zero degree of membership, order the products in ascending order on degree of membership.

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

The user agent A, computes the degree of importance of the products using the recommendation lists. The products are then suggested to human user using the following method:

1. First identify the distinct products from the lists and then compute the degree of importance (DoI) of every product (P_i) as follows:

$$DoI_i = DoT(R_1) * \{\mu_i(R_1) - \nu_i(R_1) * \pi(R_1)\} * Rank_i(R_1) \cap DoT(R_2) * \{\mu_i(R_2) - \nu_i(R_2) * \pi_i(R_2)\} * Rank_i(R_2) \cap .... \cap DoT(R_k) * \{\mu_i(R_k) - \nu_i(R_k) * \pi_i(R_k)\} * Rank_i(R_k) \quad (4)$$

where,
$DoI_i(A)$ is degree of importance of P_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 P_i according to X,
$\nu_i(X)$ is the degree of non-membership of P_i according to X,
$\pi_i(X)$ is degree of uncertainty or hesitation of P_i according to X,
$X \in \{R_j \mid j = 1, …, k\}$,
$DoT(R_j)$ is the degree of trust of the A on R_j,
$Rank_i(R_j)$ is the normalized position of P_i in the recommendation list of R_j,
k is the total number of recommenders who have recommended P_i.

2. Compute the threshold, TDOI for degree of importance as

$$TDOI = \mu - \nu * \pi \quad (5)$$

where,
μ, ν and π are degree of membership, non membership and uncertainty, respectively that the user agent expects from the interesting products.

3. For all the distinct products, P_i of step 1
   if $DOI_i < 0$ or $TDOI < DOI_i$ then
   P_i is recommended to the human user corresponding to the user agent
   The DOI_i is negative for those products that do not conform to the user taste exactly. They have been recommended as they have mass appeal or it has matched only the uncertain list and not the preference list.

4. Initializing and Updating Degree of Trust on the recommenders

4.1 Trust Initialization

When a new agent comes up in the system or the system starts from the scratch, then the agents have to initialize the trust values for some of the other agents in the application domain to form its acquaintance set. If an agent is known to the other agent (i.e., the corresponding humans know each other), then the human associated with the agent can initialize the degree of trust according to the personal dealings with the person. However, the system also allows an agent to initialize degree of trust on an agent X, on the basis of the experiences of the other agents with X, i.e., to what extent the other agents in the application domain have received good recommendations from X. The degree of trust is then regularly updated on the basis of the personal experience of the agent with X.

The new agent Y, asks for the experience of other agents' w.r.t. X. Let q agents return their experience values as the

number of good recommendations received to the total number of the recommendations received from X. Let j^{th} agent gives the experience as e_j. Then the degree of trust on X is as following:

$$DoT(X) = \frac{\sum_{j=1}^{q} e_j}{q} \quad (6)$$

where, DoT(X) is degree of trust as computed by Y on X.

If q is large, then basically we are interested in finding what is experience of the majority of the agents for which experiences can be clustered and then degree of trust be computed [Kaur et al., 2005].

4.2 Updating Trust

The degree of trust on a recommender is updated on the basis of the distance between degree of importance of the product as it is there in the aggregated list of the user agent (A) and the recommendation list of the recommender (R). The difference of opinion between the user and the recommender is computed as follows:

$$d = \frac{(D_1 + D_2 + \ldots + D_P)}{p} \quad (7)$$

where,

$D_i = \{\mu_i(R) - \nu_i(R) * \pi_i(R)\} - \{\mu - \nu * \pi\}$,

μ, ν, π; and $\mu_i(R)$, $\nu_i(R)$, $\pi_i(R)$ are as defined in the section 3.2,

p is the total number of products 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:

$$DoT(R) = DoT(R) + (d_t - d) \quad (8)$$

In our model, hence trust increases for those who give good recommendations and vice-versa.

5. Case Study

A case study of tourism recommender system is chosen to generate the recommendations for the selection of destination, travel agents and the flight schedule. The generated recommendations are compared with the manual recommendations provided by peers. The experiment is conducted to recommend cities of United States of America to the persons living in India, who want to visit USA for tourism purpose.

5.1 Setup

In this experiment, the dataset for developing the destination temporal ontology is taken from the Places Rated Almanac, by Richard Boyer and David Savageau, copyrighted and published by Rand McNally. The data set rates 329 cities of USA on the nine criteria viz. Climate & Terrain, Housing, Health Care & Environment, Crime, Transportation, Education, The Arts, Recreation, and Economics. Except for Housing and Crime criteria, higher score is better. The destination ontology is developed using Protégé 2000 with continent, country, state and city classes in the hierarchy and then extended to temporal ontology using frame and slot versioning. The ontology contains 329 cities of USA as individuals of class city but it can be populated with the individuals of the countries, states and cities of the world.

The travel agents use flight schedule temporal ontology to recommend flight information to the user agent. The flight class has five properties: Airline rating, Fare, Time taken to reach destination, Ticket class and Availability of tickets according to traveling plan, on the basis of which the tour plan is recommended. Presently, we have populated individuals of classes in the ontology manually but it can be integrated with the flight reservation systems for creating individuals dynamically.

5.2 Experiment

The system starts with five agents, each of which can act as user or recommender agent, and five travel agents that can recommend flights for the specified destinations. The recommenders suggest destinations and the travel agents to the user agent. The destinations are suggested using destination temporal ontology. The recommenders suggest travel agents on the basis of trust on travel agents. The architecture of the system enables the recommenders to compute the trust on the travel agents for the first time and then update it with the growing experience about them. The stored trust is then used to generate the recommendations about travel agents. After selection of the destination and the travel agents, the user agent seeks recommendations about the flight schedule for the selected destination(s) from the selected travel agent(s). The selected travel agent(s) uses the flight schedule temporal ontology to recommend flight schedule to the user agent.

For manual recommendations, two forms are designed: one for selecting a destination and the other for flight schedule to be filled by known travel agents. Twenty five known persons staying in USA were selected as recommenders. Five trusted travel agents were selected to provide traveling schedule on the basis of five attributes considered in designing flight ontology. The travel agents then suggested flights for the selected destination.

The recommendations for destination were taken against ten preference lists corresponding to ten different users. The persons who require these recommendations are of different age groups and different income groups, so their preferences are different. In case of manual recommendations, for every preference list, five recommenders suggest five destinations and every recommender suggest destinations for two preference lists. In system generated recommendations, all five agents suggest five destinations each for ten preferences. The results of the comparison of the system generated recommendations against the manual recommendations are shown in Fig. 2.

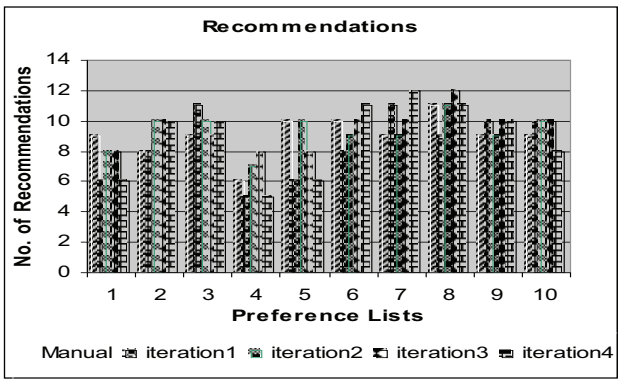

Fig. 2: Manual recommendations Vs System generated recommendations

The graph of Fig. 2 shows that the number of recommendations received from friends and system, which are above threshold of the user, follow similar pattern. Similarly, the flight schedules recommended by the system travel agents and human travel agents matches w.r.t. the cost, choice of airlines and other parameters.

The temporal effect of ontologies is under study as the ontologies have not evolved much.

6 Conclusions

Use of temporal ontologies make recommender systems independent of the knowledge base creation and this allows them to work seamlessly across different versions of ontologies. The system developer has more resources to concentrate on recommendation models without worrying about the knowledge base maintenance and evolution. The presented system based on temporal ontologies and trust network, generates the recommendations across multiple domains. The similarity measures are taken care in the form of trust update process. Intuitionistic Fuzzy Sets (IFS) have been used to capture uncertainty, inherent in the recommendation process. The tourism domain being dependent on a number of domains like air travel, geography, food, entertainment, sports etc. is taken as a landmark study. At present the case study uses two domains viz. destination and travel and can be enhanced easily for other domains.

References

[Atanassov, 1999] K. Atanassov. Intuitionistic Fuzzy Sets: Theory and Applications, *Studies in Fuzziness and Soft Computing*, Vol. 35, September 1999, Physica-Verlag.

[Bedi et al., 2006] Punam Bedi and Harmeet Kaur. Trust based Personalized Recommender System. *INFOCOM Journal of Computer Science*, 5(1):19-26, March 2006.

[Bedi et al., 2005] Punam Bedi and Sudeep Marwaha, Framework for Ontology Based Expert Systems: Disease & Pests Identification in Crops - A Case Study. In the Proc. of the International Conf. of Artificial Intelligence (IC-AI), pages 256-259, 2005.

[Bedi et al., 1993] Punam Bedi, K.D. Sharma, Saroj Kaushik. Time Dimension to Frame Systems. *Journal of Information Science and Technology*, 2(3): 212-228, April 1993.

[Bonhard et al., 2006] P. Bonhard, C. Harries, J. McCarthy and M.A. Sasse, 2006. *Accounting for Taste: Using Profile Similarity to Improve Recommender Systems*. In the *Proc. of the Conf. on Computer-Human Interaction,* Montréal, Québec, Canada, pp. 1057-1066.

[Donovan et al., 2005] John O'Donovan and Barry Smyth. Trust in Recommender Systems. In *Proc. of the International Conf. on Intelligent User Interfaces*, pages 167-174, San Diego, California, USA, January 2005.

[Guha et al., 2004] R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of Trust and Distrust. In the *Proc. of World Wide Web*, pages 403-412, New York, USA. 2004.

[Herlocker et al., 2000] J.L. Herlocker, J.A. Konstan, A. Borchers and J. Riedl. Explaining Collaborative Filtering Recommendations. In the *Proc. of the ACM 2000 Conf. on Computer Supported Cooperative Work*, Philadelphia, PA, pages 241-250, 2000.

[Karypis, 2001] George Karypis. Evaluation of Item-Based Top-N Recommendation Algorithms. In the *Proc. of the tenth International Conf. on Information and Knowledge Management*, New York, USA, 2001, ACM Press.

[Kaur et al., 2005] Harmeet Kaur and Punam Bedi. Using Fuzzy Clustering to Determine Agent Reputation. In *Proc of International Conference on Data Mining*, pages 79-85, Las Vegas, USA, June 2005.

[Massa et al., 2004] Paolo Massa, Bobby Bhattacharjee. Using Trust in Recommender Systems: an Experimental Analysis. In the *Proc. of iTrust*, pages 221-235, Oxford, UK, Springer, Vol 2995, 2004.

[Middleton et al., 2002] Stuart E. Middleton, Harith Alani and David Roure. Exploiting Synergy Between Ontologies and Recommender Systems. In *Proc. of Semantic Web Workshop*, Hawaii, USA, May 2002.

[Noy et al., 2004] Natalya F. Noy and Mark A. Musen. Ontology Versioning in an Ontology Management Framework, *Intelligent Systems*, IEEE, 19(4): 6-13, 2004.

[Sinha et al., 2001] Rashmi Sinha and Kirsten Swearingen. Comparing Recommendation made by Online Systems and Friends. In the *Proc. of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries*, Ireland, 200.

[Ziegler et al., 2004] Nicolas Ziegler, Georg Lausen. Analyzing Correlation between Trust and User Similarity in Online Communities. In the *Proc. of iTrust, Oxford*, pages 221-235, UK, Springer, Vol. 2995, 2004.

View publication stats