Exploiting Semantic Web Technologies for Recommender Systems A Multi View Recommendation Engine

Houda OUFAIDA, Omar NOUALI

DTISI Laboratory, CERIST Research Center
03, Rue frères Aissou - Ben Aknoun – Algiers, Algeria
{houfaida, onouali}@mail.cerist.dz

Abstract

Collaborative filtering systems are probably the most known recommendation techniques in the recommender systems field. They have been deployed in many commercial and academic applications. However, these systems still have some limitations such as cold start and sparsty problems. Recently, exploiting semantic web technologies such as social recommendations and semantic resources have been investigated. We propose a multi view recommendation engine integrating, in addition of the collaborative recommendations, social and semantic recommendations. Three different hybridization strategies to combine different types of recommendations are also proposed. Finally, an empirical study was conducted to verify our proposition.

Introduction

Dealing with information overload is one of the most challenging problems in the information access field; the Web is a perfect example. Unlike retrieval systems (Google, AltaVista, Yahoo, ….) which succeed in selecting suitable items according to a specific user query, these items are the same for every user in every situation, recommender systems aim to make personalized recommendation to users according to their preferences, tastes and interests expressed by users themselves or learned by the recommender system over the time.

There has been much work in this research area, from the early 1990 and still remains up to now. Foltz and Dumais experiences (Foltz and Dumais 1992) on four recommendation techniques have shown ambitious results, Resnick and collaborators proposed one of the first and probably the most known recommender system in the literature; Grouplens (Resnick et al. 1994) which recommends films to users according to their previous ratings.

Since, several models were proposed in the literature and much more applications were developed in the industry. Examples of such applications include e-commerce websites like Amazon.com for recommending books, CDs and different other items. MovieLens and Netflix for recommending movies and DVDs…

Recently, a new generation called semantic and social recommender systems have emerged taking advantage of the advancements in the semantic web technologies and features such as ontologies, taxonomies, social networks, tagging.

In this paper, we introduce a multi view recommender system that includes collaborative, social and semantic views of the user's profile. Each view recommends a set of items. Hence, three hybridization strategies are proposed for recommendations re-ranking. Finally, results from our experimentations are presented.

The rest of the paper is organized as follows: First we present the introduction of new Web 2.0 aspects in recommender systems. Then we expose our multi view recommender system, we present user's multi view representation and then present three recommendation modules: collaborative, social and semantic matching, hybridization strategies are also exposed. Finally, we discuss our experimental results and conclude with a summary of conclusions and outlooks.

Related Work

The key for an efficient recommender system is better understanding of both users and items. However, traditional recommender systems consider limited data (ratings, keywords) to compute predictions and do not take into account different factors necessary to understand reasons behind a user's judgment; is it the item's content, quality, is it because a friend recommended it?… Consequently, the users' classic communities' reflects only a *global* similarity usually insufficient to describe relations connecting users and even more items.

With the emergence of the Web 2.0, advancements allowed the apparition of a new generation of recommender systems: semantic and social recommender systems.

The availability of large product taxonomies on the Web (UNSPSC, Amazon.com, ODP for example) has encouraged the use of a taxonomy based user's/item's description in recommender systems. Quickstep (Middleton, Shadbolt, and De Roure 2004) used a paper topic ontology, AKT-ontology, to extract weighted ontology topics as user's profile. (Lops, Degemmis, and Semeraro 2007) implemented k-means clustering algorithm for neighborhood generation based on semantic similarities between users. Each user's profile contains two semantic vectors; positive and negative weighted concepts extracted from Wordnet lexical database.

Mobasher and collaborators (Mobasher, Jin and Zhou 2004) propose an enhanced similarity measure which combine two measures; a semantic items' similarity and the classical rating similarity in a linear combination to perform recommendations. Moreover, (Wang and Kong 2007) calculate three similarity measures: collaborative, semantic and *demographic* similarities. An *offline* clustering algorithm is applied to reduce computation complexity.

Another promising aspect of the semantic Web is the items' tagging (Flickr, del.icio.us). Karen and collaborators (Karen, Marinho, and Schmidt-Thieme 2008) proposed to extend User × Item rating matrix with user tags as items and item tags as users. Szomszor and al. (Szomszor et al. 2007) proposed the use of collaborative tagging, also known as *folksomies,* to enrich users' profiles. Thus, each user has a tag cloud, as well as items. User's predicted interest on each tagged item can be made based on the semantic similarity between items' tags and user's tag-clouds.

The huge popularity of online social communities, such as Facebook (175 million registration), MySpace (110 million registration) has encouraged the use of user's social and personal data in recommendation process, especially in taste related domains (movies, music,).

The first idea about the way to introduce social networks in recommender system was to replace the similarity based neighborhood formation by social neighborhood (friends and friends of friends). (Sinha and Swiringen 2001) compared collaborative recommendations made by user's friends and those predicted by the system. The results showed that users prefer friends' recommendations. This can be explained by the fact that users *trust* their friends' choices.

(Groh and Ehmig 2007) conducted an empirical study to compare collaborative and social recommendations. The experiments have shown that social recommenders perform as good as the best collaborative filtering systems when data is sparse. Similarly, (Golbeck and Ziegler 2006) developed a social network website, FilmTrust, where users manage their FOAF (Friend Of A Friend Vocabulary) based profiles and used TidalTrust algorithm (Golbeck 2005) to infer trust values over the social network. The experimental results have shown that there is a strong correlation between trust relationships and profile similarities.

(Massa and Avessani 2004) presented a trust-aware recommender system named «Web of Trust» where users define a number of users they trust. This model uses the User × Item rating matrix and the User × User trust matrix and produces as an output a predicted User × Item rating matrix less sparse from the original one. Such method is particularly beneficial in new user recommendations

Proposed Approach

Seeking on greater understanding of user's choices and judgments, we propose a novel approach which introduces social and semantic levels into the recommendation process beyond the collaborative level. Hence combining collaborative recommendations with social and semantic ones is the key idea of our proposal.

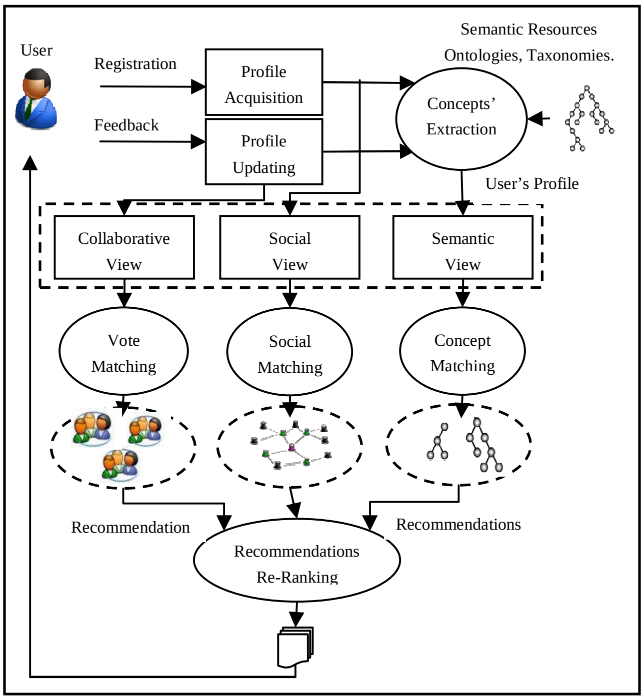


Figure 1: Multi view recommendation engine

User's Representation

Among the user's needs, user's profile is represented by three dimensions or views.

The *collaborative view* contains user's explicit or implicit ratings.

The *socio-demographic view* contains user's social data like age, gender, profession, location, personal and professional home pages, and friends' contact lists.

The *semantic view* represents user's interests in terms of a weighted concepts vector based on a hierarchical items' classification

Neighborhoods Generation

Each of the three views, proposed above, will be used by a recommendation engine to affiliate the user into a specific neighborhood and thus generate recommendations.

Collaborative Neighborhood. The collaborative view contains user's explicit or implicit ratings. Pearson Correlation can be used to compute users' similarities and k nearest neighbors' algorithm to determine such neighborhood in a classic way.

Social Neighborhood. Social recommendations are based on user's social community. It contains user's friends with trust values expressing how much the active user trusts his friends. The user annotates his relationships with such information. Trust can be binary (trust or don't trust) or on some scale, 1-5 scale where 1 is low trust and 5 is high trust. Based on these trust values, user's social neighborhood can be inferred over the social network. For example, Tidal Trust algorithm can be used (Golbeck 2006).

Semantic Neighborhood. Semantic view represents user's interests about items' content. For this, items' semantic content representation is needed.

Our choice was pointed on the use of a hierarchic semantic items' classification combined with user's evaluations to generate such view. The motivation behind this choice is the availability of such meta-information, like those of internet and e-commerce portals (Yahoo, Open Directory, LookSmart, Amazon, etc), where items are gathered into topics, which are themselves organized into a hierarchy going from the most general to the most specific.

We assume the existence of such classification H, where every item d is represented by a weighted concept vector C_d :

$$C_d = \left\{ (c_1^d, w_1^d), (c_2^d, w_2^d), \ldots (c_n^d, w_n^d) \right\}$$

The semantic view is a key element in our proposal; it is represented by weighted concepts vector C_u. These concepts are extracted from items' description C_d which the user has already rated.

$$C_u = \left\{ (c_1^u, w_1^u), (c_2^u, w_2^u), \ldots (c_m^u, w_m^u) \right\}$$

Concept's weight represents its interest score for the user. We propose the use of the weighted average to compute the concept's average rating expressing how much the user is interested in this concept; the result is divided by the maximum rating value Max_v (5 for example) to have a value between [0,1]

$$w(c) = \frac{\sum_j w_j r_{u,j}}{\sum_j w_j} \Big/ Max_r$$

User's vector C_u is updated when the active user rates a new item d. Hence, for each concept c contained in the new item's vector, there are four possible situations:

1. c already exists in C_u;
2. c is a super class concept of a concept in C_u;
3. c is a sub class concept of a concept in C_u;
4. c is a new concept, and is neither a super class nor a sub class concept of a concept in C_u;

We propose the following algorithm (Algorithm 1.) for semantic user's profile updating. It is executed for each new rating r:

Algorithm1: Profile Updating

Begin

Input $C_d = \left\{ (c_1^d, w_1^d), (c_2^d, w_2^d), \ldots (c_n^d, w_n^d) \right\}$ /* item's d *vector* */

$C_u = \left\{ (c_1^u, w_1^u), (c_2^u, w_2^u), \ldots (c_m^u, w_m^u) \right\}$ /* User's u vector */

$v_{u,d} = r$ /* user u rating on item d */

Foreach $c_i^d \in C_d \mid w_i^d \geq \min_{wd}$ **Do**

Switch c_i^d :

$c_i^d \in C_u$: /* c_i^d already exists in C_u */

$$w_{ui} = \frac{\sum_j w_j v_{uj} + w_i^d * r}{\sum_j w_j + w_i^d} \Big/ Max_v$$

/* w_i^d weight's updating*/

$\exists c_j^u \in C_u \mid c_i^d \in S(c_j^u)$ /* c_i^d super class concept of a concept in C_u*/

$C = \left\{ c' \mid c' \in C_u \ \& \ c_j^u \in S(c') \right\}$

Foreach $c' \in C$ Do

$$w_{c'} = \frac{\sum_j w_j v_{uj} + w_i^d * sim(c', c_i^d) * r}{\sum_j w_j + w_i^d * sim(c', c_i^d)} \Big/ Max_v$$

End

$\exists c' \in C_u \mid c' \in S(c_i^d)$: /* c_i^d a sub class concept of a concept c' in C_u*/

$$w_{c'} = \frac{\sum_j w_j v_{uj} + w_i^d * sim(c', c_i^d) * r}{\sum_j w_j + w_i^d * sim(c', c_i^d)} \Big/ Max_v$$

$C_u = C_u \cup \left\{ (c_i^d, \frac{w_i^d * r}{Max_v}) \right\}$ /* adding c_i^d to C_u */

Else : /*c_i^d is a new concept */

$C_u = C_u \cup \left\{ (c_i^d, \frac{w_i^d * r}{Max_v}) \right\}$ /* adding c_i^d to C_u */

End

End

End.

In order to generate recommendations based on semantic view of the user's profile, users with similar interests must be found to build semantic neighborhood.

Hierarchical concepts organization allows us to reach users with similar concepts and those having more specific concepts in their semantic views. For example, in a hierarchic film classification, if we know that a user u likes "*comedy*" films in general, he should have concept "*comedy*" with a high interest weight, "*0.9*" for example, in his semantic view and there are other users which like more specific comedy kind films such as "*dark comedy*" or "*fantasy comedy*", these users should belong to the active user's neighborhood with a certain *membership degree*. (Algorithm 2.) builds such neighborhood ;

Algorithm2 : User Concept Matching

Begin

Input $C_u = \left\{ (c_1^u, w_1^u), (c_2^u, w_2^u), \ldots (c_m^u, w_m^u) \right\}$ /* User's u vector */

Foreach $c_i^u \in C_u \mid w_i^u \geq \min_{wu}$ **Do**

$V_{init} = \left\{ u_j \mid c_i^u \in C_{uj} \right\} \cup \left\{ u_j \mid c_s \in C_{uj} \;\&\; c_s \in subconcepts(c_i^u) \right\}$

Foreach $u_j \in V_{init}$ Do

$$\deg ree(u_j) = \frac{sim(c_i^u, c_{uj})}{\left| w_i^u - w_{uj} \right| + 1}$$

Priority_List_*c_i^u.add(*u_j*,degree(*u_j*))*

End

$V_u = \left\{ \bigcup_{i=1..m} Priority\_List\_c_i^u \right\}$

End

End.

The membership degree formula is proportional to the similarity between the two users' concepts and inversely proportional to the difference between their interest scores.

Thus for each concept with a significant weight (>=min_{wu}), we look for users having the same concept in their semantic views (*Vinit*) and users with more specific concepts, *Subconcepts(c)* function looks for such users (Algorithm 3.).

Algorithm3 : SubConcepts (c)

Begin

If (*depth(c)=depth(H)*) **then** /**c* is a leaf concept*/

$subconcepts(c) = \phi$

Else

If (*depth(c)=depth(H)-1*) **then** /**c* is a super class concept of a leaf concept*/

$subconcepts(c) = \left\{ c' \mid c' \text{ IS-A } c \;\&\; \exists u \mid c' \in C_u \right\}$

Else

$subconcepts(c) = \left\{ c' \mid c' \text{ IS-A } c \;\&\; \exists u \mid c' \in C_u \right\}$

$C = \left\{ c'' \mid c'' \text{ IS-A } c' \;\&\; c' \text{ IS-A } c \right\}$

While ($subconcepts(c) = \phi$) **Do**

$subconcepts(c) = \left\{ \bigcup_{c' \in C} subconcepts(c') \right\}$

$C = \left\{ c'' \mid c'' \text{ IS-A } c' \;\&\; c' \in C \right\}$

End

End

End

End.

Once semantic neighborhood built, remains rating predictions on items (Algorithm 4.).

Recommendations' Re-Ranking

Since each collaborative, social and semantic recommendation engines produce their own list of recommendations, recommendations' re-raking is required. The question here is "which hybridization strategy to adopt?" Burke (Burke 2005) experimented five hybridization strategies: weighted, switching, cascade, feature combination and feature augmentation hybrids. In this paper, we propose three possible hybridization strategies: *mixed*, *weighted* and *switched*.

Algorithm4 : Prediction

Begin

Foreach $c_i^u \in C_u$ **Do**

While *Priority_List_*c_i^u .count > 0 **Do**

$$p_{u,j} = k \sum_{i=1}^{n} sim(u, u_i) v_{i,j}$$

with $k = \dfrac{1}{\sum_{i=1}^{n} sim(u, u_i)}$

and $sim(u, u_i) = \dfrac{1}{\sum_{i<=m} w_i^u} \sum_{i<=m} w_i^u \deg ree_i(u, u_i)$

End

End

End.

For this we introduce a confidence value per concept and per recommendation engine. This value represents how much a user likes items from a specific recommendation engine which are classified under this concept. The intuition behind this proposition is that a specific user *u* may like friends' recommendation for "*comedy*" films and semantic recommendations for "*documentary*" films for example.

Hence, for each concept in semantic view, we introduce three confidence values denoted as: F_{coll}, F_{soc} and F_{sem} for collaborative, social and semantic concept confidence. We compute the percentage of returned items that are relevant for each recommendation engine classified under a concept *c*:

$$F = \frac{\left\| \left\{ d / r_{u,d} >= R, c \in C_d, w_c >= W \right\} \right\|}{\left\| \left\{ d / c \in C_d, w_c >= W \right\} \right\|}$$

R is the minimum user's rating to be considered as relevant, 4 for example, and *W* is the minimum concept's weight in item *d* to be considered as significant, 0.7 for example.

For each concept in the semantic view, the three confidence values are maintained. Thus, the concept vector C_u is completed as follows:

$$C_u = \left\{ (c_1^u, w_1^u, p_{coll1}, p_{soc1}, p_{sem1}), \ldots (c_m^u, w_m^u, p_{collm}, p_{socm}, p_{semm}) \right\}$$

For new concepts, the three confidence values are initialized as $F_{coll} = F_{soc} = F_{sem} = 1/3$.

Mixed Hybridization. Perhaps, the first idea that comes to mind is to simply mix recommendations from the three recommendation engines. If an item is recommended from more than one engine, the final rating is calculated as the average between each engine's rating. The following linear combination computes such average:

$$r_{u,d} = \alpha . r_{coll} + \beta . r_{soc} + \delta . r_{sem}$$

With: $\alpha = \beta = \delta = 1/n$ if d is recommended by *n* recommendation engines (n<=3). If a recommendation engine doesn't recommend *d*, its corresponding rating *r* will be 0.

Weighted Hybridization. Unlike the first hybridization strategy, α, β and δ values are proportional to the confidence values of recommended item's concepts. Hence, α parameter is computed as the weighted average of item's collaborative confidence values, as well as β and δ. We propose the following algorithm to be applied to each resulting item (Algorithm 5.).

Algorithm5: Weighted Hybridization

Begin

Input $C_d = \{(c_1^d, w_1^d), (c_2^d, w_2^d), ...(c_n^d, w_n^d)\}$ /* item's *d* vector*/

$C_u = \{(c_1^u, w_1^u, p_{coll1}, p_{soc1}, p_{sem1}), ...(c_m^u, w_m^u, p_{collm}, p_{socm}, p_{semm})\}$ /* user's *u* vector*/

$$\alpha = \frac{\sum_j w_j^d p_{collj}}{\sum_j w_j^d} \quad \beta = \frac{\sum_j w_j^d p_{socj}}{\sum_j w_j^d} \quad \delta = \frac{\sum_j w_j^d p_{semj}}{\sum_j w_j^d}$$

/* $p_{collj} = p_{socj} = p_{semj} = 1/3$ **if** $c_j^d \notin C_u$ */

/* Normalization*/

$$\alpha = \frac{\alpha}{\alpha+\beta+\delta}\ ;\ \beta = \frac{\beta}{\alpha+\beta+\delta}\ ;\ \delta = \frac{\delta}{\alpha+\beta+\delta}$$

End.

Switched Hybridization. In this strategy, if an item is recommended from more than one recommendation engine, we chose the rating provided by the engine corresponding to the maximum value of item's global confidence values α, β or δ .

Experimental Evaluation

In order to experiment our multi view recommender system, we use *BookCrossing* dataset[1]. This dataset contains 42643 implicit ratings provided by 10000 users on 21944 books, which gives an average of 4.26 rating per user. These ratings were collected from *All Consuming*[2] website where people can share their interests about books, movies, food and other items. However, user's friends' list is not available, only user's age and location are available.

Amazon uses a hierarchy of nodes, called *Browse Nodes*, to organize its items for sale. Each node represents a collection of items, such as "Harry Potter books", not the items themselves. Browse nodes are related in a hierarchical structure.

Hence, for all rated books in the dataset, we crawled the Amazon web service for 15 days to get each book's nodes, the result was 309205 nodes including 6176 distinct node which gives an average of 14 nodes per book.

However, Amazon does not provide nodes' weights, for this and in order to favor most specific nodes and at the same time to diminish the weight of nodes that occur very frequently, we have estimated node's *i* weight as follows:

$$Weight(i) = \frac{\left(\frac{depth_i}{Max_{depth}} * \log\left(\frac{N}{n_i}\right)\right)}{Max_{weight}}$$

With $depth_i$ is node's *i* depth in Amazon's classification, Max_{depth} is the depth of the most specific node of the current item, *N* is number of items classified under the root node "*books*", n_i is number of items classified under node *i* and finally, Max_{depth} is used to normalize all resulting weights values for the current item. We also used *Lin* semantic similarity for this evaluation.

Our evaluation methodology was as follows. User's collaborative, social and semantic views are built. Collaborative view contains user's ratings. Since, user's friends' list data is not available; we have simulated such neighborhood by considering users living in the same location and having similar ages. For the semantic views, we have generated different user's semantic views depending on ratings number considered; seven collaborative and semantic views are constructed for each user for 1, 5, 10, 20, 30, 40, 50 ratings considered. The social view remains the same since it does not depend on user's ratings.

We have varied the number of ratings considered for the recommendation generation and then measured recommendation accuracy using MAE measure and coverage using RECALL measure, applied on each recommendation engine separately and also with mixed hybridization strategy .

For each recommendation list, we have calculated the average of MAE and Recall values for Top5, Top10, Top20, Top30, Top40 and Top50 items. Figure 2 displays our results.

Preliminary results show that in term of precision, semantic recommendation engine produce more accurate recommendations comparing it to collaborative engine, especially with small nucmber of ratings (<10) however in terms of recall, collaborative engine recommends more relevant items. Semantic engine bad recall may in part be explained by the fact that *SubConcept* function was limited at one level, i.e. we have only considered direct subclasses in user's neighborhood generation.

Mixed hybridization strategy appears to compromise between semantic recommendations good precision and collaborative recommendations good recall. It outperforms collaborative engine in terms of recall and keeps in the same time a good accuracy comparable to the semantic recommendation engine (Figure 3.).

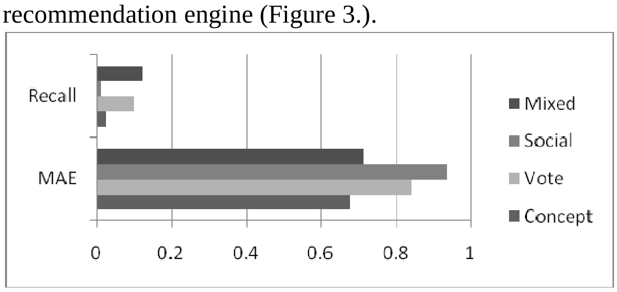
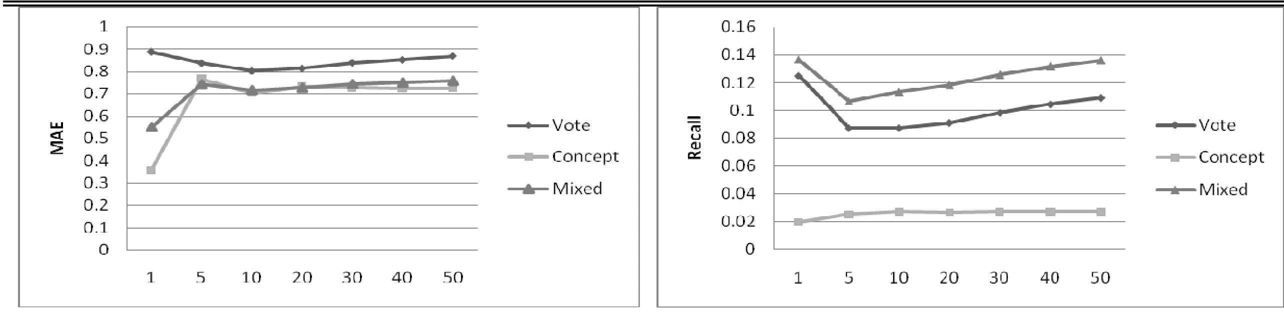


Figure 3: Comparison between collaborative, social, semantic and mixed recommendation engines

[1] http://www.informatik.uni-freiburg.de/%18cziegler

[2] http://www.allconsuming.net/

Figure 2: TopN MAE and Recall for collaborative, semantic and multi view recommendation engines



Conclusion

In this paper, we have proposed a multi view recommendation engine which exploits semantic web technologies such as semantic items' description and social networks beyond the classic ratings data. The results of our experimentations were very promising and improved the recommendation process in many ways:

1. Exploiting semantic background knowledge enriches description of different system elements (users, items);
2. Enhanced semantic description improves items' classification and users' clustering, it helps the system to produce more accurate predictions;

We believe that the introduction of a semantic level in recommender systems explains users' judgments in a semantic way and should lead to a greater understanding of the target users.

Social elements are particularly benefit in taste related domains. Our multi view recommendation system could make semantic enhanced predictions for an item's category (scientific papers for example) and social enhanced recommendations for another item's category (music, movies) if the user prefers that. Thus, experimenting this proposition in an online study will be interesting; it constitutes one possible outlook to investigate.

The use of interesting Web services which provide social data about users based on unified user's models (FOAF, APML for example) is also another interesting issue to investigate. Social communities may increase trust over recommender systems and encourage users to communicate with like-minded people. Thus, this consistent users' participation provides more information about their interests and preferences;

References

Burke, R. 2005. *Hybrid Systems for Personalized Recommendations*. Book chapter: Intelligent Techniques for Web Personalization.133-152, Springer.

Foltz, P.W. , and Dumais S.T.1992. *Personalized Information Delivery : An Analysis of Information Filtering Methods*. Communications of the ACM 35 (12), 51-60.

Golbeck, J. 2005. *Computing and Applying Trust in Web-based Social Networks*. Ph.D. thesis. University of Maryland. College Park, MD USA..

Golbeck, J., Ziegler, C.N. 2006. *Generating Predictive Movie Recommendations from Trust in Social Networks*, In Proc. of the fourth international conference on trust management.

Groh, G., and Ehmig, C. 2007. *Recommendations in taste related domains: Collaborative Filtering vs. Social Filtering*, ACM GROUP'07.

Karen, H.L.; Marinho, L.B.; and Schmidt-Thieme L. 2008. Tagaware *Recommender Systems by Fusion of Collaborative Filtering Algorithms*, ACM SAC'08.

Lops, P.; Degemmis M. ; and Semeraro, G. 2007. *Improving social filtering techniques through WordNet-Based user profiles*. UM 2007.

Massa, P., and Avesani P. 2004. *Trust-aware Collaborative Filtering for Recommender System*. In "On the Move to Meaningful Internet Systems: CoopIS, DOA, and ODBASE". Berlin, Heidelberg: Springer, pp. 3-17.

Middleton, S.E.; Shadbolt, N.R.; and De Roure, D.C. 2004. *Ontological User Profiling in Recommender Systems*. ACM Trans. Information Systems, vol. 22, no. 1, pp. 54-88.

Mobasher, B.; Jin, X.; and Zhou, Y. 2004. *Semantically enhanced collaborative filtering on the Web*. Book chapter. Web Mining: FromWeb to SemanticWeb.

Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom; P.; and Riedl, J. 1994. *GroupLens: An open architecture for collaborative filtering of netnews*. In Proc. of the 1994 Conference on Computer Supported Collaborative Work, Eds. ACM Press, New York. 175-186.

Sinha, R., and Swearingen, K. 2001. *Comparing recommendations made by online systems and friends*. DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries.

Szomszor, M.; Cattuto, C.; Alani, H.; O'Hara, K.; Baldassarri, A.; Loreto, V.; and Servedio, V.D.P. 2007. *Folksonomies, the Semantic Web, and Movie Recommendation*, In Proc. Of the ESWC'07.

Wang, R.Q., and Kong F.S. 2007. *Semantic-Enhanced Personalized Recommender System*. In Proc. of the international conference on machine learning and cybernetics.