A Multilayer Ontology-based Hybrid Recommendation Model

Iván Cantador, Alejandro Bellogín and Pablo Castells
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Campus de Cantoblanco, 28049, Madrid, Spain
E-mail: {ivan.cantador, alejandro.bellogin, pablo.castells}@uam.es

We propose a novel hybrid recommendation model in which user preferences and item features are described in terms of semantic concepts defined in domain ontologies. The concept, item and user spaces are clustered in a coordinated way, and the resulting clusters are used to find similarities among individuals at multiple semantic layers. Such layers correspond to implicit Communities of Interest, and enable enhanced recommendations.

Keywords: hybrid recommender systems, communities of interest, ontologies, user profiling

1. Introduction

Recommender systems are based on the principle that users with common traits (in their demographic data, behaviour, tastes, opinions, etc.) may enjoy similar items. However, in typical approaches, the comparison between users is done globally, in such a way that partial, but strong and useful similarities might be missed. For instance, two people may have a highly coincident taste in cinema, but a very divergent one in sports. The opinions of these people on movies could be highly valuable for each other, but risk to be ignored by many collaborative systems, because the global similarity between the users might be low.

We thus argue for the distinction of different layers within the interests and preferences of users, as a useful refinement to produce better recommendations. We propose an approach in which depending on the current context, only a specific subset of layers in a user profile is considered in order to establish the user's similarities with other people when a recommendation is to be performed. This conforms models of induced user communities, partitioned at different common semantic layers, which can be exploited in the recommendation strategies in order to produce more accurate and context-relevant results.

Our approach is based on an ontological representation of the domain of discourse where user interests are defined. The ontological space takes the shape of a semantic network of interrelated concepts, and user profiles are initially described as weighted lists measuring the users' interests for those concepts. We propose here to exploit the links between users and concepts to extract relations among users according to common interests. By analysing the structure of the ontology and taking into account the semantic preference weights of the user profiles, we cluster the domain concept space, and generate groups of interests shared by certain users. Thus, those users who share interests of a specific concept cluster are connected in the corresponding community. This conforms a richer space for analysis by the recommendation strategies where cohesiveness, focus, and outreach are explicit dimensions to the avail of enhanced content filtering methods.

The structure of the paper is as follows. Section 2 summarises the existing types of recommender systems and some of their limitations. Section 3 is dedicated to the underlying ontology-based knowledge representation and basic content retrieval of our proposal. The method to cluster the concept space in several layers of shared semantic interests is presented in section 4. The exploitation of the derived communities to enhance recommendations is described in section 5. Empirical evaluations are presented in section 6. Finally, some discussions and future research lines are given in section 7.

AI Communications
ISSN 0921-7126, IOS Press.

2. Background

The recommendation problem can be formulated as follows [1]. Let $\mathcal{U} = (u_1, \cdots, u_M)$ be the set of all users registered in the system, and let $\mathcal{I} = (i_1, \cdots, i_N)$ be the set of all possible items that can be recommended. Let $\mathcal{G}(u_m, i_n)$ be a utility function that measures the gain or usefulness of item i_n to user u_m, i.e., $\mathcal{G} : \mathcal{U} \times \mathcal{I} \to \mathcal{R}$, where $\mathcal{R}$ is a totally ordered set (e.g. non negative integers or real numbers within a certain range). Then, for each user $u_m \in U$, we aim to choose the item $i^{\max u_m} \in \mathcal{I}$ that maximises the user's utility.

$$\forall u_m \in \mathcal{U}, i^{\max u_m} = \arg \max_{i_n \in \mathcal{I}} \mathcal{G}(u_m, i_n)$$

The utility of an item is usually represented by a *rating*, measuring how much a specific user is (or is predicted to be) interested in a specific item. The utility function is defined only on the items that have been previously evaluated by the users, and it has to be extrapolated to the whole $\mathcal{U} \times \mathcal{I}$ space. Based on the mechanism in which ratings are estimated for different users, two main types of recommender systems can be distinguished: content-based and collaborative filtering systems. Due to the limitations of each of the above strategies, combinations of them have been investigated in the so-called hybrid recommender systems.

2.1. *Content-based recommender systems*

Content-based recommendation approaches [7] build on the conjecture that a person likes items with features similar to those of other items he liked in the past. Thus, the utility gain function $\mathcal{G}(u_m, i_n)$ of item $i_n \in \mathcal{I}$ for user $u_m \in \mathcal{U}$ is estimated based on the utilities $\mathcal{G}(u_m, i_l)$ assigned by user u_m to items i_l that are *similar* to item i_n.

For these techniques, several limitations have been identified in the literature [1]:

- **Restricted content analysis.** Content-based recommendations are restricted by the features that are associated with the items to be recommended. Thus, in order to have a sufficient set of features, the content should either be in a form that can be automatically parsed by a computer or in a form in which features can be manually extracted in an easy way.
- **Content overspecialisation.** Content-based systems only retrieve items that score highly against a specific user profile. They cannot recommend items that are different from anything the user has seen before.
- **Cold-start: new user problem.** A user has to rate a sufficient number of items before a content-based recommender system can really understand her preferences.
- **Portfolio effect: non diversity problem.** In certain cases, items should not be recommended if they are too similar to something the user has already seen.

2.2. *Collaborative filtering systems*

Collaborative filtering (CF) techniques [8] match people with similar preferences in order to make recommendations. The utility gain function $\mathcal{G}(u_m, i_n)$ of item $i_n \in \mathcal{I}$ for user $u_m \in \mathcal{U}$ is estimated based on the utilities $\mathcal{G}(u_l, i_n)$ assigned to item i_n by those users u_l that are *similar* to user u_m.

In CF, users express their preferences by rating items. The ratings submitted by a user are used as an approximate representation of his tastes, interests and needs. These ratings are matched against ratings submitted by all other users, obtaining the user's set of *nearest neighbours*. The items that were rated highly by the user's nearest neighbours and were not rated by the user are recommended.

Pure CF systems confront some of the weaknesses existing in content-based approaches. Since collaborative strategies make use of other users' ratings, they can deal with any kind of content and recommend any item. However, they suffer from their own limitations [1]:

- **Sparse rating problem.** The success of CF systems depends on the availability of a critical mass of users. They are based on the overlap in ratings across users and have difficulties when the space of ratings is sparse.
- **Cold-start: new user problem.** Collaborative strategies learn the users' preferences only from the ratings they have given. When a new user utilises the system none or few personal ratings are available, and no proper recommendations can be made.
- **Cold-start: new item problem.** CF systems do not make use of content information of the existing items. Until a new item is rated by a substantial number of users, a system would not be able to recommend it.

- **Gray sheep problem.** For the user whose tastes are unusual compared to the rest of the population, there will not be any other users who are particularly similar, leading to poor recommendations.

2.3. Hybrid recommender systems

Hybrid recommender systems [6] combine content-based and collaborative filtering techniques under a single framework, mitigating inherent limitations of either paradigm.

Numerous ways for combining content-based and collaborative information are conceivable [1]. Among them, the most widely adopted is the so-called *collaborative via content* paradigm, where content-based profiles are built to detect similarities among users.

Although specific weaknesses of both content-based and collaborative approaches are addressed by hybrid strategies, there still exist other general limitations in the current recommender systems.

- **No contextual information in the recommendation process.** Traditional recommender systems make suggestions based only on the user and item information, and do not take into consideration contextual information that might be crucial in some applications.
- **Non flexible recommendations.** In general, recommendation methods are inflexible in the sense that they only recommend individual items to individual users. Group recommendations are still open to innovations.
- **Scalability problem.** Nearest neighbour-based algorithms require computation that grows with the number of users and items. For them, there exist a number of dimensionality reduction techniques, such as Latent Semantic Analysis (LSA), and efficient clustering methods, such as co-clustering.

3. Ontology-based recommendations

3.1. Knowledge representation

Our approach makes use of explicit user profiles. Working within an ontology-based personalisation framework [9], user preferences are represented as vectors $\mathbf{u}_m = (u_{m,1}, \cdots, u_{m,K})$ where $u_{m,k} \in [0,1]$ measures the intensity of the interest of user $u_m \in \mathcal{U}$ for concept $c_k \in \mathcal{O}$ (a class or an instance) in a domain ontology $\mathcal{O}$, K being the total number of concepts in the ontology. Similarly, the items $d_n \in \mathcal{D}$ in the retrieval space are assumed to be annotated by vectors $\mathbf{d}_n = (d_{n,1}, \cdots, d_{n,K})$ of concept weights, in the same vector-space as user preferences.

The ontology-based representation is richer and less ambiguous than a keyword or item-based model. It provides an adequate grounding for the representation of coarse to fine-grained user interests (e.g. interest for items such as a sports team, an actor), and can be a key enabler to deal with the subtleties of user preferences. An ontology provides further formal, computer-processable meaning on the concepts (who is coaching a team, an actor's filmography).

Furthermore, ontology standards, such as RDF[1] and OWL[2], support inference mechanisms that can be used to enhance recommendations, so that, for instance, a user interested in animals (*superclass* of cat) is also recommended items about cats. Inversely, a user keen on lizards and snakes can be inferred with a certain confidence to like reptiles. Similarly, a user fascinated about the life of actors can be recommended items in which for example the name of Brad Pitt appears due to he could be an *instance* of the class Actor. Also, a user keen on Spain can be assumed to like Madrid, through the *locatedIn* relation. These characteristics are exploited in our recommendation models.

3.2. Content-based recommendation model

With the presented knowledge representation, we use a retrieval model that works in two phases. In the first one, a formal ontology-based query is issued by some form of query interface (e.g. NLP-based) formalising a user information need. The query is processed, outputting a set of ontology concepts that satisfy it. From this point, the second phase is based on an adaptation of the classic vector-space IR model, where the axes of the space are the concepts of $\mathcal{O}$, instead of keywords. The query and each item are thus represented by vectors $\mathbf{q}$ and $\mathbf{d}$, so that the satisfaction of a query by an item can be computed by its cosine measure.

The problem, of course, is how to build the $\mathbf{q}$ and $\mathbf{d}$ vectors. For more details, see [5]. Here we obviate

[1] Resource Description Framework, www.w3.org/RDF
[2] Web Ontology Language, www.w3.org/2004/OWL

this issue, and continue explaining our content retrieval process with its personalisation phase. Once a user profile is obtained, our notion of content retrieval is based on a matching algorithm that provides a personal relevance measure $pref(d,u)$ of an item d for a user u. This measure is set according to the semantic preferences of the user and the semantic annotations of the item based again on a cosine-based vector similarity $\cos(\mathbf{d},\mathbf{u})$. In order to bias the result of a search (the ranking) to the preferences of the user, the above measure has to be combined with the query-based score without personalisation $sim(d,q)$ defined previously, to produce a combined ranking.

To facilitate the matching between item and user vectors we propose a semantic preference spreading mechanism, which expands the initial set of preferences stored in user profiles through explicit semantic relations with other concepts in the ontology. Our approach is based on Constrained Spreading Activation (CSA). The expansion is self-controlled by applying a decay factor to the intensity of preference each time a relation is traversed. Thus, the system outputs ranked lists of items taking into account not only the preferences of the current user, but also a semantic spreading through the user profile and the ontology.

4. Multilayered Communities of Interest

In social communities, it is commonly accepted that people who are known to share a specific interest are likely to have additional connected interests. In fact, this assumption is the essence of the CF systems. We assume this hypothesis here as well.

A vector $\mathbf{c}_k = (c_{k,1}, \cdots, c_{k,M})$ is assigned to each concept c_k present in the preferences of at least one user, where $c_{k,m} = u_{m,k}$ is the weight of concept c_k in the semantic profile of user u_m. Based on these vectors a classic hierarchical clustering strategy is applied. The clusters obtained represent the groups of preferences (topics of interests) in the concept-user vector space shared by a significant number of users. The similarity between a user's preferences $\mathbf{u}_m = (u_{m,1}, \cdots, u_{m,K})$ and a cluster C_q is computed by:

$$\mathrm{sim}(u_m, C_q) = \frac{\sum_{c_k \in C_q} u_{m,k}}{|C_q|}$$

where c_k represents the concept that corresponds to the $u_{m,k}$ component of the user preference vector, and $|C_q|$ is the number of concepts included in the cluster.

The concept are then used to find emergent, focused semantic Communities of Interest (CoI). User profiles are partitioned into semantic segments. Each of these segments corresponds to a concept cluster and represents a subset of the user interests that is shared by the users who contributed to the clustering process. By thus introducing further structure in user profiles, it is now possible to define relations among users at different levels, obtaining a multilayered network of users.

These social networks can be exploited to the benefit of collaborative recommendations, not only because they establish similarities between users, but also because they provide powerful means to focus on semantic contexts for different information needs. The design of recommendation models in this direction is explored in next section.

5. Multilayered hybrid recommendations

Using our multilayered CoI proposal explained in the previous section, we present two recommendation models that generate ranked lists of items in different scenarios [3]. The first model (that we shall label as UP) is based on the whole semantic profile of the user to whom a unique ranked list is delivered. The second model (labelled UP-q) outputs a ranking for each semantic cluster C_q.

The two strategies are formalised next. In the following, for a user profile u_m, an information object vector d_n, and a cluster C_q, we denote by u_m^q and d_n^q the projection of the corresponding concept vectors onto cluster C_q, i.e. the k-th component of u_m^q and d_n^q are $u_{m,k}$ and $d_{n,k}$ respectively, if $c_k \in C_q$, and 0 otherwise.

Model UP

The semantic profile of a user u_m is used by the system to return a unique ranked list. The preference score of an item d_n is computed as a weighted sum of the indirect preference values based on similarities with other users in each cluster, where the sum is weighted by the similarities with the clusters, as follows:

$$\mathrm{pref}(d_n, u_m) = \sum_q \mathrm{nsim}(d_n, C_q) \sum_i \mathrm{nsim}_q(u_m, u_i)\mathrm{sim}_q(d_n, u_i)$$

The idea behind this first model is to compare the user's interests with those of the others users, and, taking into account the similarities among them, weight all their complacencies about the items. The comparisons are done for each cluster measuring the similarities between items and clusters. We thus attempt to suggest an item in a double way. First, according to the item features, and second, based on the connections among user interests, in both cases at different semantic layers.

Model UP-q
The preferences of the user are used by the system to return one ranked list per cluster, obtained from the similarities between users and items at each cluster layer. The expression is analogous to equation of model UP, but does not include the term that connects the item with each cluster C_q.

$$\mathrm{pref}_q(d_n, u_m) = \sum_l \mathrm{nsim}_q(u_m, u_l) \cdot \mathrm{sim}_q(d_n, u_l)$$

where q maximises $\mathrm{sim}(u_m, C_q)$.

Analogously to the previous model, this one makes use of the relations among the user interests, and the user satisfactions with the items. The difference here is that recommendations are done separately for each layer. If the current semantic cluster is well identified for a certain item, we expect to achieve better precision/recall results than those obtained with the overall model.

6. Experiments

Our proposal addresses some of the limitations of current recommender systems. The semantic relations between concepts and instances of the ontologies are exploited to tackle such common issues as restricted content analysis, sparsity, cold-start, content overspecialisation, or portfolio effects. Moreover, by our method for identifying multilayered CoI, we are able to discover relations between users at different levels, increasing the possibilities of finding similarities for users with unusual interests (gray sheep problem).

In this section, we report our work and results from the empirical evaluation of the hybrid models described in the previous section. Specifically, we have conducted two different experiments: one that makes use of real, manually defined user profiles, and another that exploits synthetic user profiles generated with data from the well-known IMDb[3] and MovieLens[4] datasets.

6.1. Experimenting with real user profiles

The experiment [3] was set up as follows. A set of 24 pictures was taken as the retrieval space. Each picture was annotated with semantic metadata describing what the image depicts, using a domain ontology including six topics: animals, beach, buildings, family, motor, and vegetation. A weight in $[0, 1]$ was assigned to each annotation, reflecting the relative importance of the concept in the picture. 20 graduate students of our department were asked to independently define their weighted preferences about a list of concepts, related to the above topics, occuring in semantic annotations of the pictures. No restriction was imposed on the number of topics and concepts to be selected by each of the students. Indeed, the generated user profiles showed very different characteristics, observable not only in their shared interests, but also in their complexity. Some students defined their profiles very thoroughly, while others only annotated a few concepts of interest.

6.1.1. Concept and user clustering step

Once the 20 user profiles were created, we run our method. After the execution of the preference spreading procedure, the concept space was clustered based on user interest similarities. In this phase, because our strategy is based on a hierarchical clustering, various clustering levels were found, corresponding to levels of compromise between complexity (in terms of the number of clusters) and compactness (defined by the number of concepts per cluster or the minimum distance between clusters). As a stop criterion to determine the number of clusters to be formed, a rule based on the elbow criterion was used, stating that the number of selected clusters is such that adding another cluster does not add sufficient information. A number of $Q = 4$ clusters was hereby selected.

It has to be noted that not all the clusters had assigned user profiles. However, they provide semantic relations between users, independently from their being associated to other clusters, and regardless of the number of assigned users.

[3]The Internet Movie DataBase, www.imdb.com
[4]The GrouLens research group, www.grouplens.org

Some conclusions were drawn from this experiment. Cluster 1 contained the most specific concepts related to construction and motor, showing a significant correlation between these two topics. Checking the profiles associated to the cluster we observed that, overall, they have medium-high weights on the concepts of these topics. Cluster 2 was the one with most different topics and general concepts. In fact, it was a cluster that had the weakest relations between users. Notoriously, the concepts *wife* and *husband* appeared in this cluster. These concepts were not selected by the subjects, who were students, not married at that time. Cluster 3 was the one gathering all the concepts related to beach and vegetation. The subjects who liked vegetation items also seemed to be interested in beach items. It also had many of the concepts related to the topic of animals, but in contrast to cluster 2, the annotations were for more common and domestic animals. Finally, cluster 4 collected the majority of the family concepts. It was observed that several subjects defined their preferences only in this topic.

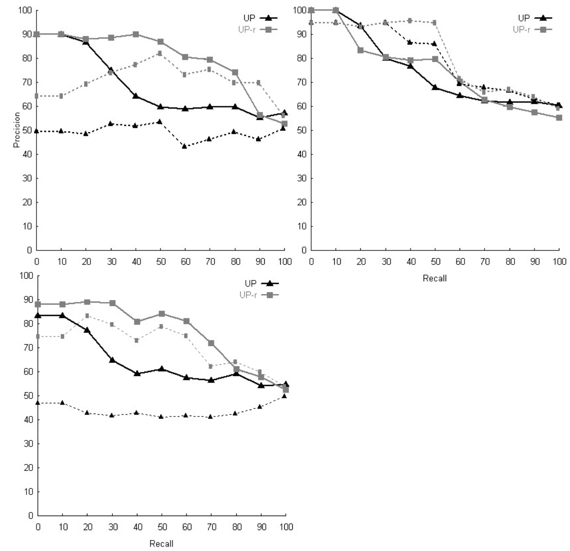

Fig. 1. Average precision vs. recall curves for users assigned to the user clusters obtained with the UP (black lines) and UP-q (gray lines) models at level $Q = 4$. The dotted lines represent the results achieved without semantic preference spreading

6.1.2. Recommendation step

We evaluated our recommendation models computing their average precision/recall curves for the users of each of the existing clusters. In this case we calculate the curves at clustering level $Q = 4$. Figure 1 exposes the results.

The version UP-q, which returns ranked lists according to specific clusters, outperforms the version UP, which generates a unique list assembling the contributions of the users in all the clusters. Obviously, the more clusters we have, the better performance is achieved. However, it can be seen that very good results are obtained with only three clusters. Additionally, for both models, we have plotted with dotted lines the curves that result without spreading user preferences. Though further, larger-scale experimentation would be in order to draw further conclusions, it can be observed that our clustering strategy performs better when it is combined with the CSA algorithm, especially in the UP-q model.

6.2. Experimenting with IMDb and MovieLens repositories

The MovieLens database, provided by the GroupLens research group, is one of the most referenced and tested repositories in the Recommender Systems research community. In its large public version, it consists of approximately 1 million ratings for $6,079$ movies by $6,040$ users on a 1-5 rating scale. The MovieLens repository is in turn based on the Internet Movie Database (IMDb), which probably constitutes the largest collection of movie-related information on the Internet. Its pages contain a catalogue of every pertinent detail about a movie, such as the cast, director, shooting locations, languages, soundtracks, etc.

In our second experiment [2], we explored the combination of both data sources. Specifically, we exploited IMDb information to produce ontology-driven, content-based user profiles from MovieLens ratings. For such purpose, we defined an ontology describing the fundamental concepts involved in IMDb, including classes such as movies, actors, directors, genres, languages, countries and keywords, and relations among them. We parsed the IMDb content, and converted it to an OWL KB, based on the aforementioned movie ontology. Semantic preferences are then built from the MovieLens ratings by means of a number of transformations exploiting the KB, which are explained in the next subsection. Table 2 shows some volumetric data about the generated dataset.

Table 1
Information about the size of the IMDb and MovieLens data and knowledge-bases used in our experiments

IMDb	Movies	$1,095,404$
	Actors	$1,451,667$
	Directors	$138,686$
	Genres	28
	Languages	295
	Keywords	$32,244$
MovieLens	Movies	$3,655$
	Users	$6,040$
	Ratings	$968,418$

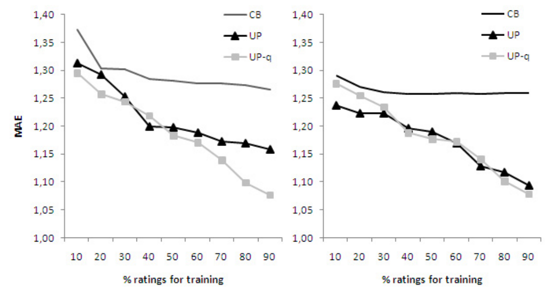

Fig. 2. MAE for our content-based (CB), UP and UP-q hybrid recommenders built with 100 (left) and 1000 (right) users, and $10\%, \cdots, 90\%$ of the available MovieLens ratings

6.2.1. Generating user profiles from MovieLens ratings and IMDb data

Let $i_{m,1}, \cdots, i_{m,N_m}$ be the N_m items (movies) rated by user u_m and let $r_{m,1}, \cdots, r_{m,N_m} \in [1,5]$ be the corresponding ratings. We define the weight of movie i_n for user u_m as $w_{m,n} = \frac{r_{m,n}}{5} \in (0,1]$.

For each user u_m we measure the relevance of the different movie features by summing the weights of the movies in which these features appear.

$$w_{m,f} = \frac{1}{N_m} \sum_{n: f \in \text{features}(i_n)} w_{m,n}$$

Taking into account all the movies rated by a user, the feature weights obtained with the previous formulas could be taken as initial user preferences. However, we noticed that a suitable part of the features should be filtered for inclusion in the final profiles, as follows. After we expanded the features, we found that some of them appeared in the user profiles with too many instances, while others with too few. According to the cumulative distributions, for each feature, we selected the number of instances which covered 90% of the feature distribution. By applying this criterion, the resulting user preferences included the 8 top-weighted genres, 3 countries, 15 actors and 3 directors per movie. On the other hand, we dismissed as user preferences the movie keywords (hundreds per movie) and the spoken languages (the majority of the movies were in English), because they were not discriminating movie features.

6.2.2. Evaluating the hybrid recommendation models

Conventional recommender algorithms are typically modelled as rating estimators. They receive a set of known ratings as input and predict new ratings for unseen items. In this context, the effectiveness of the models can be measured by the Mean Absolute Error (MAE), i.e., the mean of the absolute differences between the ratings and their predicted values.

However, since our recommendation models are defined under a personalised content retrieval framework with ranking scores ranging in $[0,1]$, and aiming to make comparisons with MovieLens ratings, our recommendations need to be mapped to 1-5 scale ratings. We tackle this based on the cumulative distributions of ratings, as follows. To normalise each predicted value $p_{m,n}$ we first map its cumulative probability $G(p_{m,n})$ to the equivalent cumulative probability $F(r_{m,n})$ in the rating value distribution. Then, we compute its inverse value $F^{-1}(G(p_{m,n}))$ to extract the corresponding rating $r_{m,n}$:

$$r_{m,n} = F^{-1}(G(p_{m,n}))$$

Once the rating transformations are defined, we can evaluate our recommenders by measuring their MAE. To this end, we built (*trained*) the models with 100 and 1000 users, taking 10% to 90% of their MovieLens ratings, leaving the remaining ratings for testing. Figure 2 shows a comparison between the MAE values obtained with the pure content-based and the hybrid recommendation models (UP and UP-q).

For both models, the obtained MAE values are not as good as they could be. It is important to note that the automatic generation of user profiles from MovieLens ratings and IMDb movie features, and the conversion of preference-based values in $[0,1]$ to 1-5 ratings, are achieved in our experiments so far by means of simple methods, leaving ample room for improvement, as they are not the focus of our research to this point. The experiments provide notwithstanding clear evidence that the cluster-oriented UP-q model appears again to be an appropriate hybrid strategy, significantly outperforming the base line established by our content-based recommender.

7. Conclusions and future work

We have presented an approach to automatically identify CoI from ontological user profiles, where the degrees of membership of the users to the communities are exploited within a multilayered hybrid recommendation model, addressing several limitations of the current recommender systems:

- **Restricted content analysis**. The use of ontologies and standard semantic technologies to describe the items to be recommended make it possible to annotate, distribute and exploit metadata from different multimedia sources, such as texts, videos or audios.
- **Content overspecialisation, cold-start, portfolio and sparsity problems**. The proposed semantic spreading method extends the user preferences and item features, enabling the detection of indirect co-occurrences of interests between users, available for use by the recommendation strategies.
- **Gray sheep problem**. The proposed hybrid model compares user profiles at different semantic layers, enabling to find focused and meaningful relations between users, reducing the gray sheep problem.

Naturally, further directions for improvement remain. For one, more efficient clustering strategies can be used in the generation of the concept clusters. We plan to explore more scalable clustering techniques based on co-clustering or dimensionality reductions, such as LSA.

Our proposal is flexible and easily portable to different applications and domains. Further enhancements can be equally explored by drawing from the achievements and ongoing work in the field of semantic-based knowledge technologies, in areas such as such as group-oriented recommendations, by combining several ontological user profiles to generate shared semantic group profiles [4], context-aware recommendations for personalised content retrieval [9], or query-driven recommendations, based on ontologies to describe item features, user preferences, and semantic queries [5].

Acknowledgements

This research was supported by the European Commission (FP6-027685 - MESH) and the Spanish Ministry of Science and Education (TIN2005-06885). The expressed content is the view of the authors but not necessarily the view of the MESH project as a whole.

References

[1] Adomavicius, G., Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17 (6), 734-749.

[2] Cantador, I., Bellogín, A., Castells, P. (2007). Modelling Ontology-based Multilayered Communities of Interest for Hybrid Recommendations. *Proceedings of the 1st International Workshop on Adaptation and Personalisation in Social Systems: Groups, Teams, Communities* (SociUM 2007). Corfu, Greece.

[3] Cantador, I., Castells, P. (2006). Multilayered Semantic Social Networks Modelling by Ontology-based User Profiles Clustering: Application to Collaborative Filtering. *Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management* (EKAW 2006), (pp. 334-349). Podebrady, Czech Republic.

[4] Cantador, I., Castells, P., Vallet, D. (2006). Enriching Group Profiles with Ontologies for Knowledge-Driven Collaborative Content Retrieval. *Proceedings of the 1st International Workshop on Semantic Technologies in Collaborative Applications* (STICA 2006), (pp. 358-363). Manchester, UK.

[5] Castells, P., Fernández, M., Vallet, D. (2007). An Adaptation of the Vector-Space Model for Ontology-based IR. *IEEE Transactions on Knowledge and Data Engineering*, 19 (2), 261-272.

[6] Melville, P., Mooney, R. J., Nagarajan, R. (2002). Content-Boosted Collaborative Filtering for Improved Recommendations. *Proceedings of the 18th National Conference on Artificial Intelligence* (AAAI 2002), (pp. 187-192). Edmonton, Canada.

[7] Mooney, R., Roy, L. (2000). Content-based Book Recommending Using Learning for Text Categorization. *Proceedings of the 5th ACM Conference on Digital Libraries*, (pp. 195-240). San Antonio, USA.

[8] Sarwar, B. M., Karypis, G., Konstan, J., Riedl, J. (2001). Item-Based Collaborative Filtering Recommendation Algorithms. *Proceedings of the 10th International World Wide Web Conference* (WWW 2001), (pp. 285-295). Hong Kong, China.

[9] Vallet, D., Castells, P., Fernández, M., Mylonas, P., Avrithis, Y. (2007). Personalized Content Retrieval in Context Using Ontological Knowledge. *IEEE Transactions on Circuits and Systems for Video Technology*, 17 (3), 336-346.