# A Multilayer Ontology-based Hybrid Recommendation Model

**Iván Cantador, Alejandro Bellogín, Pablo Castells**

Escuela Politécnica Superior
Universidad Autónoma de Madrid
Campus de Cantoblanco, 28049, Madrid, Spain
{ivan.cantador, alejandro.bellogin, pablo.castells}@uam.es

## Abstract

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 exploitation of meta-information about the recommended items and user profiles in a general, portable way, along with the capability of inferring knowledge from the relations defined in the ontologies, are the key aspects of the presented proposal. More specifically, 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 (CoI), and enable collaborative recommendations of enhanced precision. Our approach is tested in two sets of experiments: one including profiles manually defined by real users and another with automatically generated profiles based on data from the IMDb and MovieLens datasets.

**Keywords**: hybrid recommender systems, communities of interest, ontology, user profiling

## 1. Introduction

Recommender systems emerged in the early nineties as a thriving research area on its own, distinct from other related fields in Artificial Intelligence and Information Retrieval. The area has undergone a considerable leap in significance and potential value since then, with the boost of digital content and online businesses involving stocks of goods of different sorts.

The volume, growth rate, ubiquity of access, and to a large extent unstructured nature of worldwide content challenge the limits of human processing capabilities and information access technologies, putting at stake the effective utility of content, despite its actual value. It is in such settings where recommender systems can make a great valuable contribution, by proactively scanning the space of choices, and predicting the potential usefulness of items for each particular user.

Recommender systems are based on the principle that users with common traits (in their demographic data, behaviour, taste, 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 argue for the distinction of different layers within the interests and preferences of users, as a useful refinement to produce better recommendations. Depending on the current context, only a specific subset of the layers of a user profile is considered in order to establish her similarities with other people when a recommendation has to be performed. Such models of induced user communities, partitioned at different common semantic layers can be exploited in the recommendation processes in order to produce more accurate and context-sensitive 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 the 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. 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.

The rest of the paper has the following structure. Section 2 summarises the existing types of recommender systems and some of their current limitations. Section 3 is dedicated to the underlying ontology-based knowledge representation and basic content retrieval of our proposal. The mechanism 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 collaborative filtering is described in Section 5. The empirical evaluation of that model is presented in Section 6. Finally, we conclude with some discussions and future research lines in Section 7.

## 2. Background

The recommendation problem can be formulated as follows [1]. Let $\mathcal{U}=(u_1, u_2, \ldots, u_M)$ be the set of all users

registered in the system, and let $\mathcal{I} = \left( i_1, i_2, \ldots, i_N \right)$ be the set of all possible items that can be recommended. Let $g\left( u_m, i_n \right)$ be a utility function that measures the gain or usefulness of item $i_n$ to user $u_m$, i.e., $g : \mathcal{U} \times \mathcal{I} \to \mathcal{R}$, where $\mathcal{R}$ is a totally ordered set (e.g. non negative integers or real numbers within a certain range). Then, for each user $u_m \in \mathcal{U}$, we aim to choose the item $i^{\max,u_m} \in \mathcal{I}$ that maximises the user's utility. More formally:

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

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. Each element of the user space $\mathcal{U}$ can be described with a *profile* that might include several demographic characteristics, such as gender, age, nationality, etc., or some information about the user's tastes, interests and preferences. Analogously, each element of the item space $\mathcal{I}$ can be described with a set of characteristics. For example, in a movie recommender system, movies can be described not only by their titles, but also by their genres, principal actors, directors, etc.

The main difficulty lies in that the utility function $g$ is usually not defined in the entire $\mathcal{U} \times \mathcal{I}$ space. In recommender systems, the utility function is defined only on the items that have been previously rated by the users, and it has to be extrapolated to the whole $\mathcal{U} \times \mathcal{I}$ space. Based on the mechanism in which item ratings are estimated for different users, the following two main types of recommender systems can be distinguished: 1) content-based systems, in which the user is recommended items similar to those he preferred in the past, and, 2) collaborative filtering systems, in which the user is recommended items that people with similar tastes and preferences liked in the past. 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 approaches to recommendation making [14][16] 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 $g\left( u_m, i_n \right)$ of item $i_n \in \mathcal{I}$ for user $u_m \in \mathcal{U}$ is estimated based on the utilities $g\left( u_m, i_l \right)$ assigned by user $u_m$ to items $i_l$ that are "similar" to item $i_n$. For instance, in order to suggest movies to user $u_m$ a content-based system would try to understand the commonalities among movies user $u_m$ has previously evaluated positively: specific genres, preferred actors, etc.

More formally, and following the notation used in [1], let $Content\left( i_n \right)$ be the content description of item $i_n \in \mathcal{I}$, i.e., the set of content features characterising $i_n$ that are used to determine the appropriateness of the item for the different users. This description is usually represented as a vector of real numbers (weights), in which each component measures the "importance" (or "informativeness") of the corresponding feature in the item content description:

$$Content\left( i_n \right) = \mathbf{i}_n = \left( i_{n,1}, i_{n,2}, \ldots, i_{n,K} \right) \in \mathbb{R}^K$$

Analogously, let $ContentBasedUserProfile\left( u_m \right)$ be the content-based preferences of $u_m \in \mathcal{U}$, i.e., the weighted content features that describe the tastes and interests of the user.

$$ContentBasedUserProfile\left( u_m \right) = \mathbf{u}_m = \left( u_{m,1}, u_{m,2}, \ldots, u_{m,K} \right) \in \mathbb{R}^K$$

The utility gain of item $i_n$ for user $u_m$ is then calculated with a score function that combines the different item description and user profile components:

$$g\left( u_m, i_n \right) = score\left( ContentBasedUserProfile\left( u_m \right), Content\left( i_n \right) \right) \in \mathcal{R}$$

For these techniques, several limitations have been identified in the literature [1][3]:

- **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 his preferences and present reliable recommendations.

- **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 recommender systems

Collaborative filtering (CF) techniques [11][17] match people with similar preferences in order to make recommendations. In other words, the utility gain function $g\left( u_m, i_n \right)$ of item $i_n \in \mathcal{I}$ for user $u_m \in \mathcal{U}$ is estimated based on the utilities $g\left( u_l, i_n \right)$ 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 will finally be recommended. The way in which the user's "neighbours" are determined, and the strategy followed to combine the ratings of such users differentiate the existent CF approaches.

With the above ideas, the definitions of user profile and item description given in this section for content-based systems differ from those associated to CF systems. Let $CollaborativeUserProfile\left( u_m \right) = \mathbf{r}_m = \left( r_{m,1}, r_{m,2}, \ldots, r_{m,N} \right) \in \mathcal{R}^N$ be the

collaborative profile of user $u_m$ constituted by the set of ratings provided by the user to the $N$ items stored in the system, and let $Ratings(i_n) = \mathbf{r}_n = (r_{1,n}, r_{2,n}, ..., r_{M,n}) \in \mathcal{R}^M$ be the set of ratings $r_{m,n} \in \mathcal{R}$ assigned to item $i_n$ by the $M$ users registered in the system. The utility gain of item $i_n$ for user $u_m$ is then computed by a score function that combines the different user profile and item description components:

$$g(u_m, i_n) = score(CollaborativeUserProfile(u_m), Ratings(i_n)) \in \mathcal{R}$$

Pure collaborative filtering 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, even the ones that are dissimilar to those seen in the past. However, collaborative techniques suffer from their own limitations [1][3], as described next.

- **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, i.e., few users have rated the same items.

- **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 no personal ratings are available for her, 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 [3][13] combine content-based and collaborative filtering techniques under a single framework, mitigating inherent limitations of either paradigm. Thus, hybrid recommendations are generated taking into account both descriptive features and ratings.

Numerous ways for combining content-based and collaborative filtering 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 recommendation 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 their suggestions based only on the user and item information, and do not take into consideration additional 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 [15] are still open to investigation and innovations.

- **Non support for multi-criteria ratings**. Most of the current recommenders deal with single criterion ratings. However, it might be important to provide aggregated recommendations in some applications.

- **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 Singular Value Decomposition (SVD) [12], and efficient clustering methods, such as co-clustering [10].

## 3. Ontology-based recommendations

### 3.1. Knowledge representation

Our approach makes use of explicit user profiles. Working within an ontology-based personalisation framework [18], user preferences are represented as vectors $\mathbf{u}_m = (u_{m,1}, u_{m,2}, ..., u_{m,K})$ 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 $\mathbf{d_n} = (d_{n,1}, d_{n,2}, ..., d_{n,K})$ 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.

---

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

## 3.2. Content-based recommendation model

With the presented knowledge representation, we use a retrieval model (component 'Item retrieval' in Figure 1) 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 [2], 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.
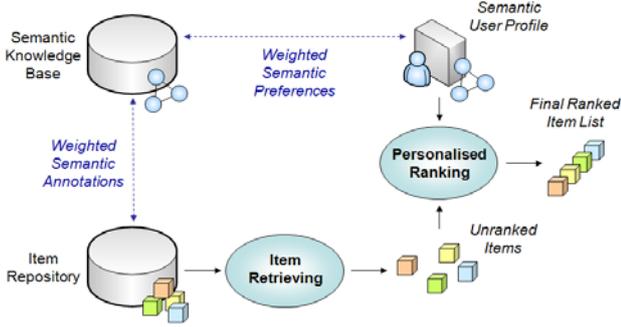


**Figure 1** *Personalised ontology-based content retrieval*

The problem, of course, is how to build the $\mathbf{q}$ and $\mathbf{d}$ vectors. For more details, see [7]. Here we obviate this issue, and continue explaining our content retrieval process with its personalisation phase (component 'Personalised Ranking' in Figure 1). 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 [18].

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) [8]. 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.

We have conducted several experiments showing that the preference extension is not only important for the performance of individual personalisation, but it is essential for the clustering strategy described in the next section.

## 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, in order to cluster the concept space in groups of preferences shared by several users.

Taking advantage of the relations between concepts and the preferences of users for the concepts, we propose to cluster the semantic space based on the correlation of concepts appearing in the preferences of individual users. After this, user profiles are partitioned by projecting the clusters into the set of preferences of each user. Then, users can be compared on the basis of the resulting subsets of interests, in such a way that several, rather than just one, weighted links can be found between two users.

Specifically, a vector $\mathbf{c}_k = (c_{k,1}, c_{k,2}, ..., c_{k,M})$ 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 [9] 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. Once the concept clusters are created, each user is assigned to a specific cluster. The similarity between a user's preferences $\mathbf{u}_m = (u_{m,1}, u_{m,2}, ..., u_{m,K})$ and a cluster $C_q$ is computed by:

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

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 clusters with highest similarities are then assigned to the users, thus creating groups of users with shared interests.

The concept and user clusters are then used to find emergent, focused semantic 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. Figure 2 illustrates this idea.
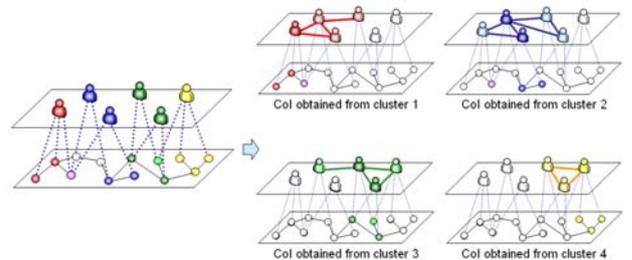


**Figure 2** *Multilayered semantic Communities of Interest built from the obtained clusters*

The figure represents a situation where four user clusters are obtained. Based on them, user profiles are partitioned in four semantic layers. On each layer, weighted relations among users are derived, building up different CoI, which 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 two recommendation models in this direction is explored in next section.

## 5. Multilayered hybrid recommendations

Using our semantic multilayered CoI proposal explained in the previous section, we present two recommendation models that generate ranked lists of items in different scenarios [5]. 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:

$$pref\left(d_n, u_m\right) = \sum_q nsim\left(d_n, C_q\right) \sum_i nsim_q\left(u_m, u_i\right) \cdot sim_q\left(d_n, u_i\right)$$

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 concept cluster measuring the similarities between items and clusters. We thus attempt to suggest an item in a double way. First, according to the item characteristics, 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 ranking that corresponds to the cluster for which the user has the highest membership value is selected. The expression is analogous to equation of model UP, but does not include the term that connects the item with each cluster $C_q$.

$$pref_q\left(d_n, u_m\right) = \sum_l nsim_q\left(u_m, u_l\right) \cdot sim_q\left(d_n, u_l\right)$$

where $q$ maximises $sim\left(u_m, C_q\right)$.

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 reduce the impact of problems such as restricted content analysis, sparsity, cold-start, content overspecialisation, or portfolio effects. Moreover, through our mechanism for identifying multilayered CoI, we are able to discover relations between users at different levels, augmenting the possibilities of finding similarities for those users with unusual interests (gray sheep problem). On the other hand, our user profile representation and content retrieval mechanism are open to new strategies for group-oriented, context-aware and query-driven recommendations, research fields which we have already started to investigate.

In this section, we evaluate empirically the hybrid models explained in the previous section. Specifically, we distinguish two different experiments: one that makes use of manually defined real user profiles, and other that exploit synthetic user profiles generated with data from the well-known IMDb[3] and MovieLens[4] repositories.

### 6.1. Experimenting with real user profiles
The experiment [5] was setup as follows. A set of 24 pictures was considered 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 and existing in the picture semantic annotations. 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 joint interests, but also in their complexity. Some students defined their profiles very thoroughly, while others only annotated a few concepts of interest. This fact was obviously very appropriate for the experiment done. In a real scenario where an automatic preference learning algorithm should be used, the obtained user profiles would include noisy and incomplete components that would hinder the clustering and recommendation mechanisms.

---

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

**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 according to similar user interests. In this phase, because our strategy is based on a hierarchical clustering, various clustering levels were found, expressing different compromises between complexity, described in terms of number of clusters, and compactness, defined by the number of concepts per cluster or the minimum distance between clusters. A stop criterion had then to be applied in order to determine the number of clusters to be chosen. We used a rule based on the elbow criterion [9], which says we should choose a number of clusters so that adding another cluster does not add sufficient information. We finally selected $Q = 4$ clusters.

It has to be noted that not all the clusters had assigned user profiles. However, they provided semantic relations between users, independently of being associated to other clusters or the number of assigned users. Table 1 shows the clusters obtained at level $Q = 4$. We have underlined those general concepts that initially did not appear in the profiles and were in the upper levels of the ontology. Inferred from our preference spreading strategy, these concepts do not necessary define the specific semantics of the clusters, but help to build the latter during the clustering processes.

| Cluster | Concepts |
|---|---|
| *1* | **ANIMALS**: Rabbit<br>**BUILDINGS**: Construction, Speedway, Racing-Circuit, Short-Oval, Garage, Lighthouse, Pier, Beach-Hut, Mountain-Shelter, Mountain-Villa, Mountain-Hut,<br>**MOTOR**: Vehicle, Ambulance, Racing-Car, Tractor, Canoe, Surf, Windsurf, Water-Motor, Sleigh, Snow-Cat, Lift, Chair-Lift, Toboggan, Cable-Car |
| *2* | **ANIMALS**: Organism, Agentive-Physical-Object, Reptile, Snake, Tortoise, Sheep, Dove, Fish, Mountain-Goat, Reindeer<br>**BUILDINGS**: Non-Agentive-Physical-Object, Geological-Object, Ground, Artefact, Fortress, Road, Street<br>**FAMILY**: Civil-Status, Wife, Husband<br>**MOTOR**: Conveyance, Bicycle, Motorcycle, Helicopter, Boat |
| *3* | **ANIMALS**: Animal, Vertebrates, Invertebrates, Terrestrial, Mammals, Dog, 'Tobby' (instance of Dog), Cat, Horse, Bird, Eagle, Parrot, Pigeon, Butterfly, Crab<br>**BEACH**: Water, Sand, Sky<br>**VEGETATION**: Vegetation, 'Tree' (instance of Vegetation), 'Plant' (instance of Vegetation), 'Flower' (instance of Vegetation) |
| *4* | **FAMILY**: Family, Grandmother, Grandfather, Parent, Mother, Father, Sister, Brother, Daughter, Son, Mother-In-Law, Father-In-Law, Cousin, Nephew, Widow, 'Fred' (instance of Parent), 'Christina' (instance of Sister), 'Peter' (instance of Brother) |

*Table 1 Concept clusters obtained at clustering level Q=4*

Some conclusions can be drawn from this experiment. Cluster 1 contains the most specific concepts related to construction and motor, showing a significant correlation between these two topics. Checking the profiles of the users associated to the cluster, we observed they overall have medium-high weights on the concepts of these topics. Cluster 2 is the one with more different topics and general concepts. In fact, it was a cluster that did have the most

weakness relations between users. It is also notorious that the concepts 'wife' and 'husband' appear in this cluster. This is due to these concepts were not be by the subjects, who were students, not married at that moment. Cluster 3 is the one that gathers all the concepts about beach and vegetation. The subjects who liked vegetation items also seemed to be interested in beach items. It also has many of the concepts belonging to the topic of animals, but in contrast to cluster 2, the annotations were for more common and domestic animals. Finally, cluster 4 collects the majority of the family concepts. It could be observed that several subjects only defined their preferences in this topic.

**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 3 exposes the results.
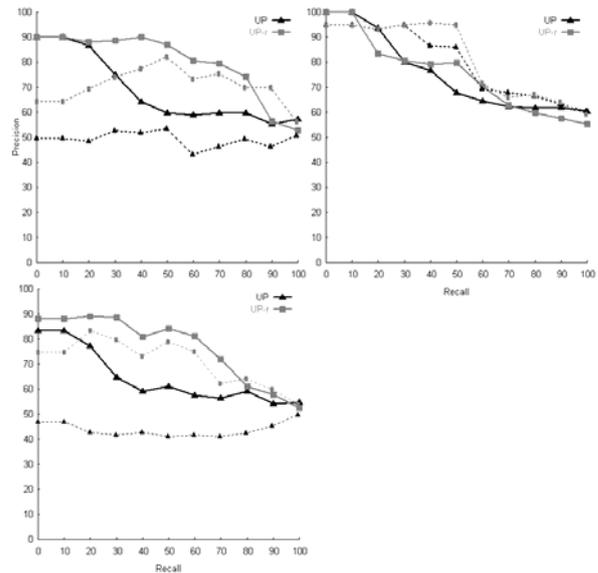


*Figure 3 Average precision vs. recall curves for users assigned to the user clusters obtained with the UP (black lines) and UP-q (gray lines) models at level Q=4. The dotted lines represent the results achieved without semantic preference spreading*

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 achieved without spreading the user preferences. Although more statistically significant experiments have to be done in order to make founded conclusions, it can be pointed out that our clustering strategy performs better when it is combined with the CSA algorithm, especially in the UP-*q* model. This fact let give us preliminary evidences of the importance of spreading the user profiles before the clustering processes.

## 6.2. Experimenting with IMDb and MovieLens repositories

The MovieLens database, provided by the GroupLens research group, is one of the most referenced and evaluated 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 [4], we explore the combination of both sources of data. Specifically, we exploit some of the IMDb information to produce ontology-driven, content-based user profiles from MovieLens ratings. For such purpose, we have 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 have 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 gathers information about the size of the data and KB generated.

| IMDb | Movies | 1,095,404 | | MovieLens subset | Movies | 3,655 |
|---|---|---|---|---|---|---|
| | Actors | 1,451,667 | | | Users | 6,040 |
| | Directors | 138,686 | | | Ratings | 968,418 |
| | Genres | 28 | | | | |
| | Languages | 295 | | | | |
| | Keywords | 32,244 | | | | |
| IMDb OWL KB | Statements | 79,689,194 | | | | |
| | Classes | 115 | | | | |
| | Disk space | ~40GB | | | | |

*Table 2* *Information about the size of the IMDb and MovieLens data and knowledge-bases used in our experiments*

Once the ontology and user profiles were built, we evaluated our hybrid recommendation models, comparing them against our pure content-based recommendation algorithm and a classic collaborative filtering strategy.

**Generating user profiles from MovieLens ratings and IMDb data**

Let $i_{m,1}, i_{m,2}, ..., i_{m,N_m}$ be the $N_m$ items (movies) rated by user $u_m$ and let $r_{m,1}, r_{m,2}, ..., r_{m,N_m} \in [1,5]$ 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.

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

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 we had to filter and select an appropriate proportion of the features to be included 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 very few. For instance, we observed that in general the initial user profiles contained lots of keywords and very few directors (Figure 4).
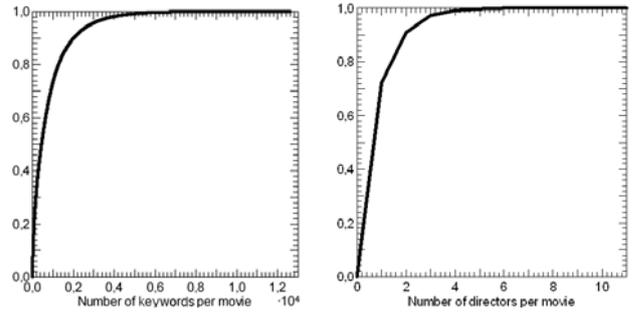


*Figure 4* Cumulative distributions of IMDb features (genres, keywords, languages, countries, actors, directors) per movie

According to the cumulative distributions, for each feature, we selected the number of instances which covers 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 rejected as user preferences the movie keywords (hundreds per movie) and the spoken languages (the majority of the movies were in English).

**Evaluating the hybrid recommendation models**

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

$$MAE = \frac{1}{M} \sum_{m=1}^{M} \frac{1}{N_m} \sum_{n=1}^{N_m} |r_{m,n} - p_{m,n}|$$

However, since our recommendation models are defined under a personalised content retrieval framework that generates rankings with values in [0,1], and aiming to make comparisons with MovieLens ratings, we need to convert our recommendations into 1-5 scale ratings. To tackle this issue, we use the rating cumulative distributions. In Figure 5, we show the distributions $F$ and $G$ of real MovieLens ratings and values obtained with our recommenders.
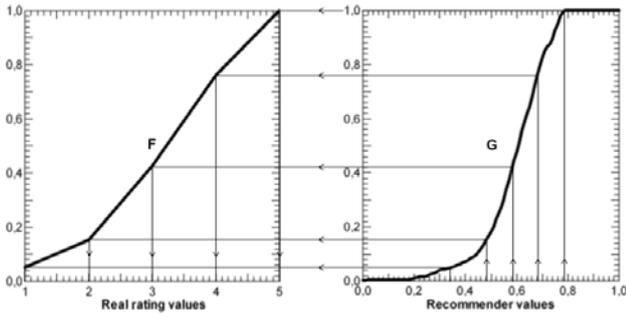
**Figure 5** *Cumulative distributions mappings of our recommender values into MovieLens ratings*

To normalise each predicted value $p_{m,n}$ we first map its cumulative probability $G(p_{m,n})$ into the equivalent cumulative probability $F(r_{m,n})$ in the rating value distribution. Then, we calculate 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 are able to evaluate our recommenders by measuring their MAE. To this end, we built ("trained") the models with 100 and 1000 users and considering 10% to 90% of their MovieLens ratings. The rest of their ratings were used for testing. Figure 6 shows a comparison between the MAE values obtained with the pure content-based and the hybrid recommendation models (UP and UP-*q*).
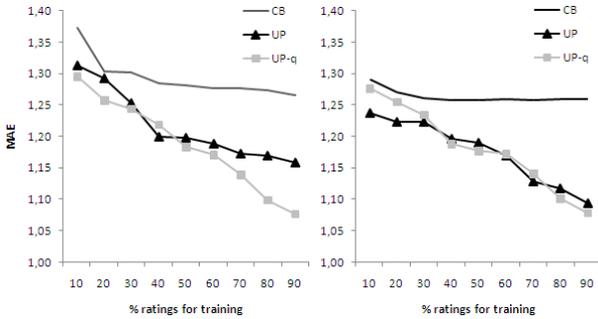


**Figure 6** *MAE for our content-based (CB), UP and UP-q hybrid recommenders built with 100 (left) and 1000 (right) users, and 10%,…, 90% of the available MovieLens ratings*

For both models, the obtained MAE values are not as good as they could be. It is very important to note that the way in which the user profiles are generated from MovieLens ratings and IMDb movie features, and the mechanism performed to convert [0,1] personalisation values into 1-5 ratings, are, without any doubt, processes which can be improved. However, this was not the purpose of our experiment. The important conclusion here is 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.

Apart from the comparison between our content-based and hybrid models, we also wanted to investigate the behaviour of classic CF when few ratings are available. Using a public implementation[5] of the item-based CF algorithm, we measured its MAE on the previously used rating datasets. Figure 7 shows the results for the CF and UP-*q* approaches. When less than the half of the available ratings were used for building the models, our recommender outperformed the collaborative approach, demonstrating thus that the former might be useful when no many ratings are available, and might successfully confront the cold-start and sparsity problems.
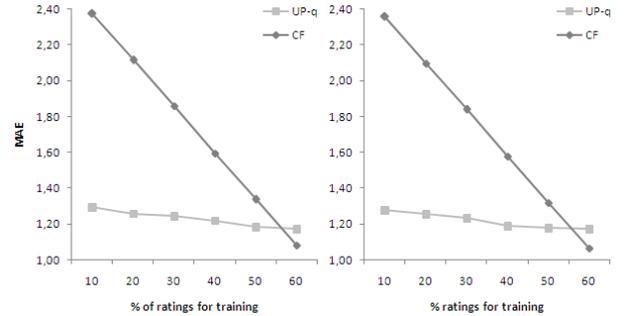


**Figure 7** *MAE for UP-q and a classic collaborative filtering recommenders built with 100 (left) and 1000 (right) users*

## 7. Conclusions and Future Work

We have presented an approach to automatically identify Communities of Interest (CoI) from ontology-based user profiles, where the degrees of membership of the users to the communities are exploited within an enhanced, 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 semantic spreading mechanism extends the user preferences and item features, facilitating the detection of indirect co-occurrences of interests between users, and promoting new interests during the recommendation processes.

- Gray sheep problem: The proposed hybrid model compares user profiles at different semantic interest layers, enabling further opportunities to find relations between users, reducing the gray sheep problem.

Naturally, further directions for improvement remain. For example, we are aware of the need of an efficient clustering strategy to generate our concept and user

---

[5] Taste Java library, http://taste.sourceforge.net

clusters. In the first version of our model, we have adapted a classic hierarchical, but we have planned to implement a more scalable clustering technique based on co-clustering [10] or dimensionality reductions, such as LSA [12].

The proposed approach is flexible and easily portable to different applications and domains. Further enhancements can be explored drawing from the achievements and ongoing work in the field of semantic-based knowledge technologies, such as:

- Group-oriented recommendations. We have studied strategies which combine several ontology-based user profiles to generate a shared semantic profile for a group of users [6].

- Context-aware recommendations. Under our ontology-based knowledge representation, we have defined the notion of runtime semantic context, and applied it for personalised content retrieval tasks [18].

- Query-driven recommendations. The use of ontologies to describe the item features and user preferences have allowed us to apply semantic search mechanisms [7] based on Semantic Web standards of query languages, such as SPARQL[6].

## Acknowledgements

## 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] Baeza-Yates, R., Ribeiro Neto, B. (1999). *Modern Information Retrieval*. Harlow, UK: Addison-Wesley.

[3] Balabanovic, M., Shoham, Y. (1997). Fab: Content-Based Collaborative Recommendation. *Communications of the ACM archive*, 40 (3), 66-72.

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

[5] 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.

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

[7] 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.

[8] Cohen, P. R., Kjeldsen, R. (1987). Information Retrieval by Constrained Spreading Activation in Semantic Networks. *Information Processing and Management*, 23 (4), 255-268.

[9] Duda, R. O., Hart, P., Stork, D. G. (2001). *Pattern Classification*. NY, USA: John Wiley.

[10] George, T., Merugu, S. (2005). A Scalable Collaborative Filtering Framework based on Co-Clustering. *Proc. of the 5th IEEE Conf. on Data Mining* (ICDM 2005), (pp. 625-628). Houston, USA.

[11] Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., Riedl, J. (1997). GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, 40 (3), 77-87.

[12] Landauer, T. K., Foltz, P. W., Laham, D. (1998). Introduction to Latent Semantic Analysis. *Discourse Processes*, 259-284.

[13] 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.

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

[15] O'Connor, M., Cosley, D., Konstan, J. A., Riedl, J. (2001). PolyLens: A Recommender System for Groups of Users. *Proc. of the 7th European Conf. on Computer Supported Cooperative Work* (ECSCW 2001), (pp. 199-218). Bonn, Germany.

[16] Pazzani, M. J., Billsus, D. (1997). Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning*, 27(3), 313-331.

[17] 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.

[18] 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.

---

[6] SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/