# RECOMMENDATION SYSTEM IN AN AUDIOVISUAL DELIVERY PLATFORM

José Mª Quinteiro-González, Ernestina Martel-Jordán, Pablo Hernández-Morera
Aaron López-Rodríguez, Leidia Martel-Monagas
*IUMA Sistemas de Información y Comunicaciones-Tecnología de la Información*
*Universidad de Las Palmas de Gran Canaria, Campus Universitario de Tafira, s/n, Las Palmas de Gran Canaria, Spain*

Angelo Santana-del Pino
*Departamento de Matemáticas, Universidad de Las Palmas de Gran Canaria, Campus Universitario de Tafira, s/n*
*Las Palmas de Gran Canaria, Spain*



Keywords: Recommendation Systems, Ontology, Customization.

Abstract: One of the main tasks of the information services is to help users to find information that satisfies their preferences reducing their search effort. Recommendation systems filter information and only show the most preferred items. Ontologies are fundamental elements of the Semantic Web and have been exploited to build more accurate and personalized recommendations by inferencing missing user preferences. With catalogs changing continuously ontologies must be built autonomously without expert intervention. In this paper we present an audiovisual recommendation engine which uses an enhanced ontology filtering technique to recommend audiovisual content. Experimental results show that the improvements of the ontology filtering technique generate accurate recommendations.

## 1 INTRODUCTION

With the Internet people are becoming overwhelmed by information. Recommender systems seem to be the choice to help users in the situation of filtering information based on user preferences.

One of the most widely used recommendation technique is *collaborative filtering* (CF) (Herlocker et al., 1999). This technique recommends items based on the experience of the user as well as other similar users. Unfortunately, CF suffers from sparsity and cold-start (Sarwar et al., 2001) (Vozalis and Margaritis, 2003). *Sparsity* refers to the lack of user ratings. This reduces the probability of finding users who have rated the same items. Clustering techniques partially solve these problems at the expense of losing accuracy in the recommendations (O´Connor and Herlocker, 1999). The *cold-start* problem is a consequence of the unconstrained search space which requires many items to be rated in order to find the right neighbours.

Another recommendation technique is the *content-based filtering* (Pazzani and Billsus, 2007). where the user will be recommended with items similar to the ones the user preferred in the past. This kind of recommendation technique does not suffer from the cold-start problem. However, this recommendation technique always ends up recommending items which have already been rated by the user (*overspecialization*).

An alternative to solve the above-mentioned problems is to use the *ontology filtering* technique (Schickel-Zuber, 2007). This technique infers user ratings using a mechanism over an ontology of the domain.

In this paper we use the ontology filtering technique on an audiovisual delivery platform. This recommendation technique has been exploited to customize both audiovisual content and advertisements according to the user profile and their behavior in the platform.

Quinteiro-González J., Martel-Jordán E., Hernández-Morera P., López-Rodríguez A., Martel-Monagas L. and Santana-del Pino A..
RECOMMENDATION SYSTEM IN AN AUDIOVISUAL DELIVERY PLATFORM.
DOI: 10.5220/0003063103790382
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2010), pages 379-382
ISBN: 978-989-8425-28-7

# 2 BACKGROUND

In this section we present the scenario where the recommendation engine has been developed, and the ontology filtering technique.

## 2.1 Audiovisual Platform

The audiovisual platform architecture is basically formed by repositories, managers and a user interface. This architecture follows the SOA (*Services Oriented Architecture*) model where each manager provides its functionality through web services.

The *user manager* module is in charge of user authentication as well as the maintenance of user session and identification through the platform.

The *content manager* module manages the audiovisual content resources.

The *virtual character manager* manages the appearance and behavior of virtual character that interact with the user.

The *statistic manager* module is in charge of maintaining the key performance indicators.

The *customization manager* provides the most adequate audiovisual contents to users regarding their taste and preferences.

The *advertisement manager* is in charge of managing all the data regarding advertisements within the audiovisual platform.

## 2.2 Ontology Filtering

*Ontology filtering* (Schickel-Zuber, 2007) solves the problems of other recommendations techniques by means of a ontology containing the item domain, and an inference mechanism which allows inferring user preferences using the proximity to other items of the ontology which have been rated by the user.

A node in the ontology represents a concept. An item is an instance of a concept, and the edges represent features that differentiate the set of sub-concepts of the same parent.

For each concept *c*, two properties are added: the *score* (*S*(*c*)) and the *a-priori score* (*APS*(*c*)). The former represents how much a specific user likes a given concept, whereas the latter determines the propagation parameters of user's scores between the sub-concepts and the concept *c*, considering only the location of concept *c* in the ontology.

To predict user ratings of a concept *y*, the system uses knowledge of the user ratings, *S*(*x*) associated to the closest concept *x*, and the *lowest common ancestor* to concepts *x* and *y* in the ontology.

Recommendations are computed by finding unseen items that are instances of concepts with the highest score.

Changes in the set of items are incorporated in the catalog, through the use of clustering algorithms to generate ontologies from the set of user ratings. The clustering algorithms used are the agglomerative and the partitional. The *criteria functions* used to optimize the allocation of an item to a given cluster are shown in Table 1.

Table 1: *Criteria functions* used by clustering algorithms.

| | | |
|---|---|---|
| 1 | $\mathcal{I}_1$ | $maximize \sum_{r=1}^{k} \frac{1}{n_r} \left( \sum_{i,j \in C_r} sim(i,j) \right)$ |
| 2 | $\mathcal{I}_2$ | $maximize \sum_{r=1}^{k} \sum_{i \in C_r} sim(i, C_r^t)$ |
| 3 | $\mathcal{E}_1$ | $minimize \sum_{r=1}^{k} n_r sim(C_r^t, C)$ |
| 4 | $\mathcal{G}_1$ | $minimize \sum_{r=1}^{k} \frac{cut(C_r, C-C_r)}{\sum_{i,j \in C_r} sim(i,j)}$ |
| 5 | $\mathcal{H}_1$ | $maximize \frac{\mathcal{I}_1}{\mathcal{E}_1}$ |
| 6 | $\mathcal{H}_2$ | $maximize \frac{\mathcal{I}_2}{\mathcal{E}_1}$ |
| 7 | $slink$ | $\max_{i \in C_i, j \in C_j} sim(i,j)$ |
| 8 | $clink$ | $\min_{i \in C_i, j \in C_j} sim(i,j)$ |
| 9 | $UPGMA$ | $maximise \frac{1}{n_i n_j} \sum_{i \in C_i, j \in C_j} sim(i,j)$ |

where *k* is the amount of cluster to consider, $C_r$ represents cluster *r*, $n_r$ denotes the amount of elements in cluster *r*, $C_r^t$ is the centroid of cluster *r*, and *sim*(*i*,*j*) is the similarity between items *i* and *j*.

Combining the *criteria functions* in Table 1 and the clustering algorithms, the system generates 15 different ontologies (agglomerative with *criteria function* 1 to 9, partitional with *criteria function* 1 to 6). Instead of using the same ontology for all users, the best ontology is selected based on the user preference profile in order to increase the recommendation accuracy.

# 3 RECOMMENDATION ENGINE ARCHITECTURE

The recommendation engine has a layered architecture, where the inner layer contains the agents that implement different filtering techniques. Above this layer, the recommendation agent drivers let you configure what techniques to use and how to combine them, and to determine the set of item ratings, product catalogs and ontologies to be used.

The architecture allows us to add new personalization techniques easily by implementing the corresponding agent. This goal has been achieved by means of software architecture presented in Figure 1. The decision tree chooses the recommendation technique which is carried out by

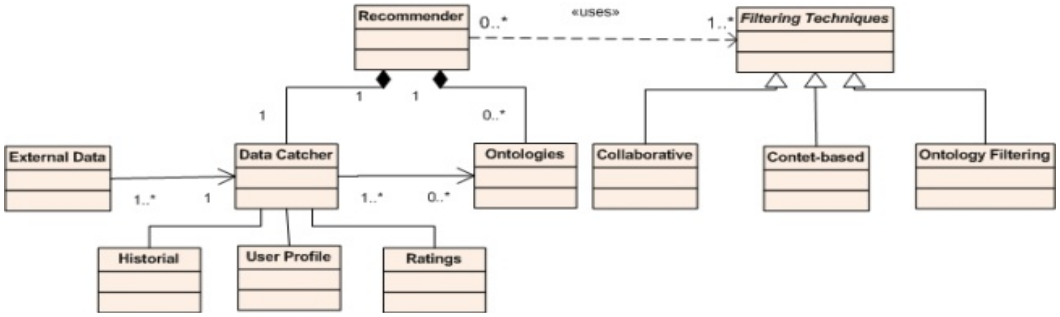


Figure 2: Class hierarchy for recommender.

the recommender. The recommender sets up the interaction between agents and data management. This interaction allows agents to get the required data for providing recommnedation results.

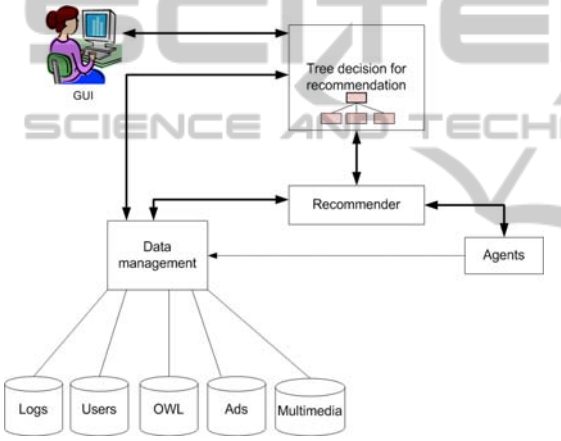


Figure 1: Software architecture for the application layer.

The class hierarchy which implements the recommender is presented in Figure 2 where filtering techniques have been separated from the data required for recommendations. Each new agent requires adding a new class for the recommender, another for the filtering technique and the necessary data classes for such a filtering technique.

The agent that implements the ontology filtering technique has two components: the ontology generator and the recommendation generator. The *ontology generator* computes off-line the set of ontologies specified in subsection 2.2. The *recommendation generator* selects the most appropriate ontology for each user.

Based on the selected ontology, the user ratings and the product catalog, the recommendation generator performs the necessary tasks to generate the recommendations.

# 4 EXPERIMENTAL RESULTS

We have been working with an improved version of the ontology filtering technique, by estimating the missing values in the user-item matrix by means of an imputation method. The algorithm of the *k* nearest neighbours has been used as the imputation method of missing values (Troyanskaya et al., 2001).

To evaluate the recommendation engine with *knn* imputation values, several experiments have been performed using the data from the MovieLens data set (Dataset MovieLens, 2010). 250 users from this dataset have been randomly selected.

The set of ratings of each user is randomly split into a training set and a test set, so that the former contains 90% of the ratings.

The training set is used to generate the ontologies. 15 ontologies have been generated by combining agglomerative and partitional methods with each *criteria function* listed in Table 1. Similarly, another 15 ontologies have been generated but from an array of user-item ratings with data imputation. All of this leads to a total of 30 ontologies.

For each of the ontologies generated, the inferred scores are compared with the real scores of those concepts in the test set, computing the NMAE (*Normalized Mean Absolute Error*) (Herlocker et al., 2004).

Then, the ontology with the lowest average error is selected. The results are shown in Table 2, which shows the number of times a clustering method with a specific *criteria function* has been selected.

Table 2 shows that more than half, namely 134 users are benefiting from the use of *knn* data imputation.

The mean error using the *ontology filtering* method is 0.056, and in the case of using *ontology filtering* with *knn* data imputation the error is 0.048.

Table 2: Distribution of selected ontology for each user.

| | | *Clustering method* | | | |
|---|---|---|---|---|---|
| | | *agglom.* | *agglom-knn* | *part.* | *part-knn* |
| *Criteria function* | *1* | 13 | 16 | 12 | 0 |
| | *2* | 6 | 8 | 4 | 4 |
| | *3* | 4 | 9 | 3 | 2 |
| | *4* | 9 | 3 | 0 | 11 |
| | *5* | 11 | 14 | 9 | 5 |
| | *6* | 3 | 8 | 2 | 2 |
| | *7* | 22 | 12 | NA | NA |
| | *8* | 8 | 25 | NA | NA |
| | *9* | 10 | 15 | NA | NA |
| **Total** | | 86 | 110 | 30 | 24 |

In relative terms this means an improvement of 14.1%.

To determine if this difference between the average errors of both methods is significant we have used the Wilcoxon test for paired samples, since there is no normal distribution of errors, as can be seen in Figure 3.

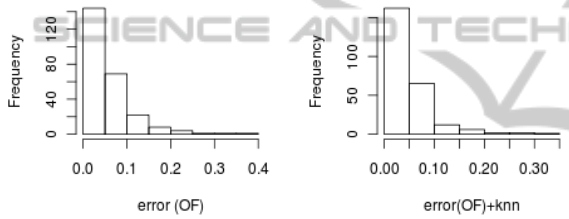


Figure 3: Wilcoxon test.

The *p*-value Wilcoxon test is less than $10^{-6}$ which indicates that the difference between the magnitudes of the errors is significant and cannot be attributed to chance.

# 5 CONCLUSIONS

In this work we have presented the recommendation engine developed for the customization and advertisement managers within an audiovisual content delivery platform. The main novelty of our work is the reusable approach of the recommendation engine for different contexts. In this sense, we can add new different filtering techniques and different content types easily. In this work we have used the ontology filtering technique.

This work has enhanced the traditional ontology filtering technique by means of an imputation of ratings in the user-item matrix.

# 6 FUTURE WORK

The extended use of social networks introduces an important element to extend user profile as the user interacts with other users in the network. Moreover, some geolocalization aspects must be considered for recommending advertisements.

At the moment we are updating the ontology filtering technique in order to improve the quality of the recommendations even more.

# REFERENCES

Herlocker, J., Konstan, J.A., Borchers, A, Riedl, J., 1999. An Algorithmic FrameWork for Performing Collaborative Filtering. In *22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, pp. 230-237.

Sarwar, B., Karypis, G., Konstan, J., Riedl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *10th Conference on World Wide Web*, Hong Kong, pp. 285-295.

Vozalis, E., Margaritis, K.G. 2003. Analysis of Recommender Systems' Algorithms. In *6th Hellenic European Conference on Computer Mathematics and its Applications (HERCMA)*, Athens, Greece.

O´Connor, M., Herlocker, J. 1999. Clustering Items for Collaborative Filtering. In *Proceedings of the ACM SIGIR Workshop on Recommender Systems: Algorithms and Evaluation*, Berkeley, CA.

Pazzani, M.J., Billsus, D. 2007. Content-Based Recommendation Systems. *Lectures Notes in Computer Science*, vol. 4321, Springer, pp. 325-341.

Schickel-Zuber, V. 2007. *Ontology Filtering*. Thesis, Lausanne, EPFL.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., Altman, R.B. 2001. Missing value estimation methods for DNA microarrays. *Bioinformatics*, vol. 17, n. 6, pp. 520-525.

Dataset Movielens. http://www.cs.umn.edu/Research/GroupLens/data (visited March 23, 2010).

Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T. 2004. Evaluating Collaborative Filtering Recommender Systems. In *ACM Transactions on Information Systems*, vol. 22, n. 1, pp. 5-53.