A Linguistic Framework for Collaborative and Knowledge-based Filtering: How to Refine Collaborative Filtering Recommendations

M.J. Barranco, L.G. Pérez, L. Martínez

Department of Computer Sciences, University of Jaén,
Jaén, 23071, Spain
E-mails: barranco, lgonzaga, martin@ujaen.es
www.ujaen.es

Collaborative filtering tecniques turned out to be one of the key elements in the success of e-commerce companies. These tools reduce the information overload that customers suffer when they purchase items on the internet. However, the constant and rapid growing of the Internet, both in number of users and in offered items, has showed that collaborative filtering techniques need to be improved. For instance, when people visit some sites, such as FilmAffinity, the recommendations provided are very diversified. This might be a problem if users are looking for items with specific features and none of them satifies them. In this contribution we present a hybrid recommender system to overcome this drawback. Our proposal uses a knowledge-based recommender system to refines the set of items recommended leaving out those ones that do not match with the current user necessities.

Keywords: Linguistic Framework, Hybrid Recommender Systems, Collaborative Filtering, Knowledge-Based Filtering

1. Introduction

Information overload is one of the main problem users face on the Internet. Search engines were developed to assist users in finding information, web portals to gather the most common services that could be needed and recommender systems to lead customers to the most suitable items for them.

In order to implement a recommender system, several techniques can be employed. Depending on the technique used, recommender systems can be classified in: Demographic,[7] Content-based,[9] Collaborative Filtering,[4] Knowledge-based[2] and Hybrid Recommender Systems.[1]

The most widespread type of recommerder system is the collaborative filtering one since it has been successful in many situations.[10,14] However, as commercial sites keep growing, both in users and in items offered, new drawbacks in their use arose. For instance, because of the huge amount of users and/or items presented, to guarantee an adequate performance, new collaborative filtering techniques were developed such as the model-based algorithms.[15] Other problematic situations are to infer recommendations to new users. In such cases, the system should be improved to overcome the cold-start problem.[13]

In this contribution we focus on a problem that users may suffer visiting some web sites as, for example, FilmAffinity (www.filmaffinity.com). Collaborative Filtering Recommender System recommends, for a specific user, the N items with the greatest predicted preference value among a huge number of potential alternatives. These recommendations are inferred without gathering any information about the items. Thus, if a user is looking for a item, a movie in FilmAffinity, with some particular features, it may be possible that none of them satisfy those expectations. To overcome this drawback, Filmaffinity offers a very basic and limited user interface that let users state some of desired features. However, we think the use of Knowledge Based Recommender System to filter the potential recommendations could provided some important advantages and more accurate results.

In this proposal, we present a hybrid recommender system that use a knowledge based one to filter the results infered by the collaborative filtering one. In the following section we will review the two recommendation algorithms used in this proposal. In section 3 the hybrid model is presented and finally some conclusions and future works are pointed out.

2. Preliminaries

In this section we will explain the two models used in our proposal. First of all, we will give a brief review of the collaborative filtering algorithm used in FilmAffinity and finally, the knowledge-based algorithm will be presented.

2.1. *Collaborative Filtering Algorithm used by FilmAffinity*

Not only is the site FilmAffinity (http://www.filmaffinity.com) one of the largest movie fan community sites in which people can receive information about movies, reviews, and recommendations, but it is also an example of how a collaborative recommender system can play a key role (be fundamental) in the success of a web site.

Collaborative Filtering Algorithms obtain the information needed to infer the recommendations for users, by inquiring them the rating of some items[11] or, in an implicit way, from the available data on user activity.[12] The most usual method to obtain this information, and the method used by FilmAffinity, are by means of explicit ratings.

These algorithms are based on the assumption that similarities in the past assessments tend to be kept in the future ones. Depending on the way that this information is exploited, collaborative filtering methods can be classified in: (i) the model based-approach[5] , when a model, defined from the rates provided by the users, is used to infer the recommendation and (ii) the memory-based approach[11] when the whole set of rates is used.

The algorithm used in Filmaffinity, a memory-based one, is a neighborhood-based collaborative filtering method.[6] Neighborhood-methods can be separated into three steps:(i) compute the weight of the users, according to the similarity with the target user or other requirements, (ii) define the subset of users, the closest to the target user, that will be used to infer the recommendation(s) (the items that have the greatest predicted ratings), (iii) and compute the recommendations by using a weighted combination of ratings of the selected neighbors.

2.2. *Knowledge Based Recommender Systems*

Knowledge Based Recommender Systems[2] provide recommendations based on inferences about the users' necessities. In these systems, users give an example of a their expectations, and from this item, the system infers good approximations of the features desired by them. It then recommends those items that are closer to the user's expectations

Let $A = \{a_1, a_2, ..., a_n\}$ be the set of items that can be recommended and $C = \{c_1, c_2, \ldots, c_l\}$ the set of features that describe the item, which are assessed by linguistic terms. The linguistc approach is suitable in this environment becase, normally, the information provided by the experts will be vague or imprecise and cannot be easily assessed in a quantitative form. The fuzzy linguistic approach represents qualitative aspects as linguistic values by means of linguistic variables[16] whose semantic is represented by fuzzy sets. For example, a set of seven terms T, could be given as follows:

$$T = \{t_0 : Null, t_1 : Low, t_2 : Middle, t_3 : High, t_4 : Perfect\}$$

Briefly, a knowledge-based recommender scheme consists of a four-step process used to infer the recommendations is:

(1) *Building the items' description database:* This is an off-line phase that includes descriptive information about items. Given that the knowledge degree about the item's features may be different, the

system offers several linguistic term sets with different granularities, $\{T_1, \ldots, T_k, \ldots, T_m\}$. So, each feature can be assessed in a different set. Hence, in the database, each item is described by a vector $R_i = \{r_{i,1}, \ldots, r_{i,j}, \ldots, r_{i,l}\}$, where $r_{i,j} \in T_k$ is the value for the feature c_j for the item i, assesed in a linguistic term set T_k.

(2) *Adquisition of the example selected by the user:* the user choose an example, a_k, that captures their liking.

(3) *Item filtering:* the calculation of the similarity with the selected example is accomplished as following:

(a) *Computation of the similarity in each feature:* Let a_i the item to be compare with the user's selected example a_k. Due to the multi-granularity, to compute the similarity between two labels that could belong to different term sets we propose the use of the following measure of resemblance between two fuzzy sets that was also used in [8] for similar purposes:

$$d_j\left(a_i, a_k\right) = \sup_x \min\left(f_A\left(x\right), f_B\left(x\right)\right)$$

Where A and B are the fuzzy sets that stand for the semantic of the linguistic labels $r_{i,j}$ and $r_{k,j}$.

(b) *Aggregation of the distance to obtain an overall similarity:* In this phase, different aggregation operators can be applied. The simplest one is the arithmetic mean. In such a case, the function used to computed the overall similarity between a_i and a_k is:

$$sim\left(a_i, a_k\right) = \frac{1}{l}\sum_{j=1}^{l}\left(d_j\left(a_i, a_k\right)\right)$$

being l the number of features.

(c) *Top-N filtering:* now, it filters the items providing the set A_R with the top N items according to the order given by the overall similarity.

(4) *Recommendations:* the system recommended the set A_R ordered by means of their similarity.

3. A Hybrid Model to Refine Collaborative Recommendations

Our contribution addresses the problem of information overload in the results of the collaborative filtering algorithm by mixing a collaborative system with knowledge based one. For mixing both algorithms we propose the use of the method cascade:[3] firstly, a collaborative algorithm, whose results are the input data of the knowledge based algorithm that provides recommendations closer to the particular interests of the user. The five-step process accomplished by the hybrid recommender system is (Figure 1):

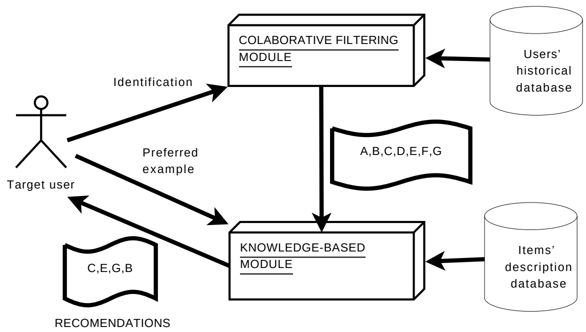

Fig. 1. A cascade-hybrid recommender system

(1) *Identification of the user*: the user should log in as the collaborative module deals with historical data and needs to know that user.
(2) *Acquiring of the preferred example*: Using classical techniques of searching, the user find an item, a_k according to their current necessities.
(3) *Collaborative filtering stage:* The collaborative module will provide a potential set of recommendations as it was shown is section 2.1.
$$A_{R_1} = \{a_i | p_{u_a,i} \geq \alpha_1\}$$
(4) *Knowledge-based filtering stage:* The input of this stage is the recommendations set provided by the collaborative stage, A_{R_1}. To accomplish this stage, the knowledge based recommender algorithm is used, as it is explained in section 2.2, to refine the recommendations sets according to example, a_k. A new set of items A_{R_2} is then obtained, including the top N products of A_{R_1} most similar to the product a_k.
(5) *Recomendacion:* Finally, the recommendations provided to the user are:

$$A_R = (a_i | a_i \in A_{R_2}, sim(a_i, a_k) \geq sim(a_j, a_k) \Leftrightarrow i \leq j)$$

4. Conclusions and Future works

In this contribution we have presented a hybrid recommender system. It combines a Collaborative Filtering algorithm with a Knowledge-based one to refine the recommendations inferred by the former in those cases that items' features could be taken into account. Since the information provided, by means of the Knowledge-Based Recommender System, is more accurate, the recommendations are closer to the real users expectations.

5. Acknowledgements

This work is partially supported by the Research Project TIN-2006-02121, P08-TIC-3548 and FEDER funds.

References

1. C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based information in recommendation, 1998.
2. R. Burke. Knowledge-based recommender systems. *Encyclopedia of Library and Information Systems*, 69(32), 2000.
3. R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.
4. D. Goldberg, D.Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61 – 70, 1992.
5. N. Good, J.B. Schafer, J.A. Konstan, A. Borchers, B. Sarwar, J.L. Herlocker, and J. Riedl. Combining collaborative filtering with personal agents for better recommendations, 1999.
6. J.L. Herlocker, J.A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering, 1999.
7. B. Krulwich. Lifestyle finder: intelligent user profiling using large-scale demographic data. *AI Magazine*, 18(2):37–45, 1997.
8. Luis Martínez, Luis G. Pérez, and M. Barranco. A multigranular linguistic content-based recommendation model. *International Journal of Intelligent Systems*, 22(5):419 – 434, 2007.
9. M. J. Pazzani, J. Muramatsu, and D. Billsus. Syskill webert: Identifying interesting web sites, 1996.
10. U. Shardanand. Social information filtering for music recommendation, 1994.
11. U. Shardanand and P. Maes. Social information filtering: Algorithms for automating “word of mouth”, 1995.
12. L. Terveen, W. Hill, B. Amento, D. McDonald, and J. Creter. Phoaks: a system for sharing recommendations. *Commun. ACM*, 40(3):59–62, 1997.
13. Rui-Qin Wang and Fan-Sheng Kong. Semantic-enhanced personalized recommender system, 2007.
14. Z.Q. Wang and B.Q. Feng. Collaborative filtering algorithm based on mutual information. *Advanced Web Technologies and Applications, Lecture Notes in Computer Science*, 3007:405–415, 2004.
15. J. Wolf, C. Aggarwal, K-L. Wu, and P. Yu. Horting hatches an egg: A new graph-theoretic approach to collaborative filtering, 1999.
16. L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning-i, ii, iii. *Information Sciences*, 8-9:199–249, 301–357, 43–80, 1975.