A Hybrid Movie Recommender System Based on Neural Networks

Christina Christakou and Andreas Stafylopatis
National Technical University of Athens
School of Electrical and Computer Engineering
157 80 Zographou, Athens, Greece
chris@softlab.ntua.gr, andreas@cs.ntua.gr

Abstract

Recently, there has been a lot of speculation among the members of the artificial intelligence community concerning the way AI can help with the problem of successful information search in the reservoirs of knowledge of Internet. Recommender systems provide a solution to this problem by giving individualized recommendations. Content-based and Collaborative Filtering are usually applied to predict these recommendations. A combination of the results of these two techniques is used in this work in order to construct a system that provides more precise recommendations concerning movies. The MovieLens data set was used to test the proposed hybrid system.

Keywords

Recommender, Collaborative Filtering, Content-Based Filtering, Hybrid System

1. Introduction

The rapid growth of the Internet has led to a new era of information. It is common place that during the period 2000-2004 the active Internet population increased 125.2% worldwide [9]. The World Wide Web provides a new means of communication that exceeds by far the traditional means of communication (radio, telephone, television). It has a great impact not only on the academic research but on the daily life as well. We have found ourselves in front of the revolution of the way information is collected, stored, processed, presented, shared and used. Data in the form of text, picture and video files are found to be abundant and easily accessible.

Easily accessible does not coincide with easily found, though. Users are confronted with situations where they have too many options. The enormous quantity of information that exists on the Internet could possibly make a very important part of it to fall into disuse provided that it remains unorganized and difficult to be located and used by a simple user. As a result, individuals need the assistance of experts in order to limit effectively and as fast as possible their preferences from the myriads of available possibilities.

This problem does not arise only in the case of research or job related activities, like the search of scientific articles, but also in the field of entertainment and amusement. That is why recommender systems have been developed in order to propose web pages, Netnews, restaurants and so on. These systems are based on modeling the end user preferences, the content and the social groups the user belongs to, so as to provide good recommendations [1].

We are not that far away from the time when the sector of entertainment of a "smart home" will be a whole department of its own. For instance, music, plays, books, cinema, films to buy/rent, concerts that suit the preferences of each individual will be successfully suggested. As far as movie-recommendations are concerned, the problem of selecting a nice film will get more and more intense as time passes. The e-Hollywood will eventually come up and anyone will be given the opportunity to be the director of its own virtual movie.

To address this problem we have developed a hybrid system that takes into consideration the kinds of a movie, the synopsis, the participants (actors, directors, scriptwriters) and the opinion of other users as well.

2. Related Work

The electronic means offers new opportunities in developing recommenders that adapt to the changing interests of users over time. Fab [6] is a system that recommends web pages. It is a distributed application

Proceedings of the 2005 5th International Conference on Intelligent Systems Design and Applications (ISDA'05)
0-7695-2286-06/05 $20.00 © 2005 IEEE

of a hybrid system, part of the digital libraries academic program of Stanford. The recommendation process is separated in two stages: the collection of items in order to create a data base, and then the selection of the suitable elements from the base for the individuals. During the collection stage pages are collected that are relevant to a small number of subjects, i.e. clusters of interests that have been produced electronically and follow the changing preferences of the population. These pages are then delivered to a larger number of users via the selection stage. A subject may arouse the interest of many users, and a user may be interested in a lot of subjects. User feedback is of great importance. It is stored so as not to be overridden by other users' feedback, while web pages that are highly scored are automatically directed to neighbors.

MovieMagician [5] is a hybrid system that provides a rating prediction when requested. The features of a movie (kind, actors, directors) are captured in a generic granularity hierarchy that is independent of a particular film. Any specific movie is an instantiation of this hierarchy and the degree to which the instantiation hierarchies of two movies overlap defines their similarity. Therefore, the features of a film can be used to find cliques, filter out irrelevant movies, annotate preferences about various features and generate explanations for a movie.

3. Content-Based and Collaborative Filtering

Information retrieval systems allow users to ask questions in order to select elements that suit a special subject and satisfy a particular need for information. These techniques, however, are not helpful in the real process of recommendation, since they do not conceive any information concerning the user preferences apart from the particular question.

The demographic filtering approach uses the descriptions of users in order to learn the relation between an element and the type of persons they like it. The profiles of users are created with the classification of users into stereotypes. Therefore, the system recommends the same elements to users with similar demographic features. Since each user is different, this approach is proved to be too general. Moreover, it cannot adapt to the changing interests of a user over time. However, demographic filtering is a useful technique when combined with other filtering approaches.

Content-based filtering recommends elements to the user based on the descriptions of previously evaluated items. Therefore, it recommends elements because they are similar to the items that the user has liked in the past. User profiles are created by extracting the characteristic features from these evaluated products or services. Such a system has various disadvantages, though. It is based on the objective information of the elements. Thus, it does not take into account the subjective attributes of an element like the atmosphere of a restaurant or the quality of filming. Moreover, it is limited to providing only few similar proposals and the quality of the recommendations is not acceptable unless there is enough interaction with the user. Nevertheless, these insufficiencies can be dealt with if content-based filtering is combined with collaborative filtering.

Collaborative filtering matches persons with similar interests and provides recommendations based on this matching. Suggestions are usually extracted from the statistical analysis of the patterns and correlations of elements that are explicitly estimated by different users or tacitly taken by monitoring the behavior of the users. Instead of calculating the similarity between elements, the similarity between users is calculated. The user profile consists of information provided by the user. This information is compared to that provided by other users in order to find the overlaps of interests among individuals. Therefore, a set of "k-nearest neighbors" is assigned to each person based on the correlation of previous evaluations. Predictions for the unknown elements are then made by using a combination of the nearest neighbors' results. The recommendation quality is generally high even when the user provides only few evaluations. However, this method presents certain disadvantages. The cold-start problem is encountered when a new item is added and it has no evaluations. There is always the chance of encountering users with particular preferences that are difficult to satisfy. If the number of users is actually small, the quality is low. Finally, there is a great difficulty in exploiting user feedback since the adaptation of the whole neighborhood is required.

Hybrid systems take advantage of content-based and collaborative filtering characteristics, as the two approaches are proved to be almost complementary. They come through all the constraints described above and as a result they enhance both performance and reliability. The exact way of combing the two approaches is a topic of current research and relies on the specific application [2], [3], [4], [7], [11], [12], [13], [14], [15].

4. Hybrid Filtering Approach

4.1 Data representation

The implementation of our approach was based on the MovieLens data set that was collected by the GroupLens Research Project at the University of Minnesota through the MovieLens web site [8].

The characteristics of this set are:

- 100,000 evaluations (in a scale from 1 to 5) of 1682 films by 943 users.
- Each user has evaluated at least 20 films.
- The types to which films belong are found to be in accordance with the types in the Internet Movies Data Base (www.imdb.com).
- Simple demographic information for the users (sex, profession, age) is available. However, we didn't use this information, since its contribution is not big enough to justify the increase in system complexity.

Moreover, we retrieved the synopsis of each movie and the contributors (director, actors and script writers) automatically by using a Java parser and the URLs provided by MovieLens.

The user evaluates films that he/she has seen in a scale of five degrees (5: masterpiece to 1: bad film). Films worth suggestion are considered those that receive grades 4-5, while those that receive 1-3 are rejected. This is essential in order to learn the preferences of the user and construct the user's profile. We take into consideration two elements: the content of films that the individuals have already seen and the films that persons with similar preferences have liked.

By the term "content of film" we refer to a set of elements that can determine verbally the parameters of the film. An important parameter is the kind of a film. Furthermore, the contributors of a film are taken into account, namely directors, script writers and actors. Finally, the summary of a film is also examined (with the addition of the title). Obviously, usual and unimportant words (such as conjunctions and pronouns) are removed and the frequency of words is calculated in order to determine those that are important and characterize the film.

By using solely the content, we would be restricted to propose only similar films, bore the user and make him/her gradually abandon the system. Moreover, no elements are included that could characterize the quality of the film. To eliminate this disadvantage, the opinions of other users are considered. Of course, only those are used that present particularly similar or particularly opposite preferences. We are thus led to a hybrid system based on both content and collaborative filtering.

It should be noted that, in order to achieve reduction of dimension in our experiment, we only consider contributors that take part in more than one film. Similarly, as far as the summary is concerned, we use only the words that are encountered in more than one film. A further refinement could be applied by using more elaborate text processing techniques [10]. If new films are added, the above parameters are calculated again in a simple way. Therefore, the data set consists of three main matrices with the characteristic features of each film: Kinds (19 kinds x 1682 films), Stars (4416 persons x 1682 films) and Synopsis (8595 terms x 1682 films).

4.2 Neural network-based content filtering

We have designed the content filtering part of the system by constructing three neural networks (Multi-Layer Perceptrons) for each user, which correspond to the three movie features considered above. The whole approach was implemented using the Matlab environment.

The training of each network is based on a sub-matrix of the corresponding main matrix that contains only the films that the specific user has already evaluated. The output of each network is of size 5 to simulate the scale of five degrees.

The main parameters of the neural networks, such as the transfer functions and the training scheme (the Resilient Back Propagation method is adopted) have been determined through experimentation using the Neural Network Toolbox. In order to further increase the success rate and provide an individualized system, we use a different number of neurons for the hidden layer and a different number of training epochs for each user and neural network (Kinds, Stars, Synopsis). The choice of the appropriate number is based on the success rate obtained in each case. (In case of a draw, the smallest number of neurons is selected.) To test the possible alternatives, a validation procedure is applied by splitting the data that concern the specific user into three sets: a training set, a validation set and a test set. Particular emphasis is given to avoiding overtraining.

Once the main options concerning network structure and training parameters have been determined, the actual training of the networks is performed. For the subsequent performance evaluation of the system, the data are randomly split into 2 sets, the Training Set (7/10) and the Test Set (3/10).

We consider successful evaluation of a movie one that correctly characterizes a film as worthy of recommendation or not. We do not stick to the exact correspondence with the desirable exit value. To explain this, we should remind that worthy of recommendation are considered those films that receive grades 4-5, while those that receive 1-3 are rejected. Thus, if the system estimates that a film is graded with 4, while the user has graded it with 5, we consider that it is successfully proposed. This way, we turn the 5-degree system into binary, since we aim at recommending a movie or not, while keeping all the information during training.

Finally, we keep the results of each neural network in a matrix, thus obtaining three matrices for each user representing the result of the content filtering part.

To make the whole procedure clear, we will present a simple example. Let us consider a user who has evaluated 100 movies. This means that we will use 70 movies for training and 30 movies for testing in each of the three neural networks. The Kinds neural network accepts 19 inputs and has 5 outputs, as described before. Each input represents a movie kind, like sci-fi, and its value is 1 when the movie belongs to this kind, 0 otherwise. The Stars neural network accepts 4416 inputs and has 5 outputs. Each input represents a movie star and its value is 1 when the star takes part in the movie, 0 otherwise. The Synopsis neural network accepts 8595 inputs and has 5 outputs. Each input represents a word and its value is the frequency of the word in the current movie synopsis. After the training, we have three networks ready to provide estimations. We test them over the 30 films of the testing set and keep the result of each network in a matrix (30 films x 5 outputs).

4.3 Collaborative part

Using the Pearson formula we find the correlation between the specific user and the rest of the users:

$$r(x,y)=\frac{\sum_{f\in films}(R_{x,f}-\overline{R}_x)(R_{y,f}-\overline{R}_y)}{\sqrt{\sum_{f\in films}(R_{x,f}-\overline{R}_x)^2\sum_{f\in films}((R_{y,f}-\overline{R}_y)^2}}$$

where $R_{x,f}$ is the evaluation of user x for film f, and $\overline{R}_x$ the mean value of the evaluations of user x.

The training set is the same used in the content filtering part. For each user we find the correlation with all other users for the films of the training set. We actually use the subset of the common films.

The correlation r(x,y) is a decimal number that lies within [- 1, 1]. The opinion of a user y is taken into consideration provided that he/she presents a very strong or very loose correlation with user x. When we have a big r (r > 0.4) and user y has given high grades to films of the test set (4-5), the counter of positive proposals (for the specific film) is increased. Obviously, if the grades are low (1-3), the counter of negative proposals is increased. It was also considered useful to take into account the opinion of users with small r (r < - 0.5). In this case, the counter of positive proposals is increased when the grades are low, since we consider that the preferences of users don't match. Similarly, the counter of negative proposals is increased when we have high grades. It is noted, however, that the weight given to an opinion in the case of loose correlation is half the one corresponding to a strong correlation. This is due to the fact that people generally tend to trust more the individuals with whom they share a lot of common elements.

Finally, the counters of positive and negative opinions provide the percentage by which the film is proposed (rate > 50%) or not. We keep this percentage in order to use it in the combining stage.

4.4 Combination of filtering parts

The aim of the application is to propose a number of films that will be of interest to the user. Of course, it is not possible to recommend all the movies that are supposed to be worth suggestion, because this would lead to the initial problem of many choices and could reduce the effectiveness of the system. For the above reasons, the number of recommendations is limited to 5 films.

The films that are proposed during the evaluation phase come from the test set. Since we have randomized the creation of the training and test set, it cannot be certain that there will be 5 movies in the test set that the user will like. That is why we provide 5 recommendations when there are at least 5, otherwise the maximum available number. These films constitute the recommendation set. Therefore, the performance is calculated as the percentage of recommendation set films that the user really likes.

In order to choose which 5 films to recommend, we follow a number of steps (rules) that are applied sequentially until the desired number is reached.

1. Initially, select films that are proposed by all four individual criteria – kind, contributors, summary (content criteria) and opinion of other users (collaborative criterion).
2. Add the films that satisfy the collaborative criterion using a high threshold.

3. Add the films that are proposed by all the content criteria (especially in the case of new movies, for which no evaluations are available).

4. Add the films that are proposed by exactly two content criteria plus the collaborative criterion.

5. Add the films that are proposed by exactly two content criteria.

6. Add the films that are proposed by one content criterion.

7. Finally, if movies are still missing from the recommendation set, propose the most popular films.

This way, we ensure that there are always recommendations. However, in real cases, we suggest the use of the three first rules with given weights (percentages). If the proposed movies do not suffice to create a complete recommendation set, the above seven-step procedure can be followed.

5. Experimental Results and Discussion

Comparative performance results are displayed in the following table.

Methodology	Precision (%)	Recall (%)
MovieLens	66	74
MovieMagician Feature-based	61	75
MovieMagician Clique-based	74	73
MovieMagician Hybrid	73	56
OurSystem Kinds-based	61	62
OurSystem Stars-based	59.3	62.5
OurSystem Synopsis-based	59.4	65.6
OurSystem Collaboration	72	78.5

As can be seen, the success of each content criterion alone is roughly 60%. This is due to the fact that just a single feature is not enough, as it can be shared by lots of different films. For example, many films may include towers and dragons and can be totally different, such as terror, sci-fi or love. The same thing applies to the film contributors. A film with extraterrestrial beings by Steven Spielberg cannot be usually compared to an ordinary film with aliens.

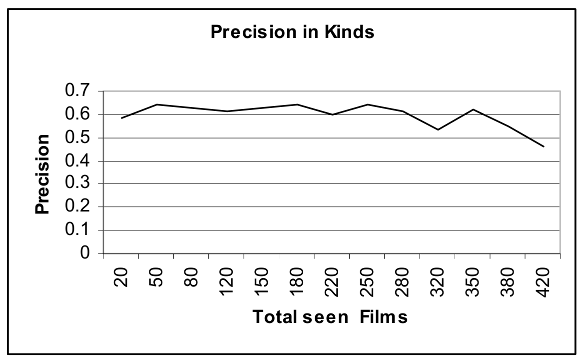
For this reason, the combination of the different characteristics is justified. We have already pointed out that we are not interested in proposing all those films that the user would probably like, since this would lead us back to the problem of infinite choices. A more effective approach is to suggest a few films with increased probability of success.

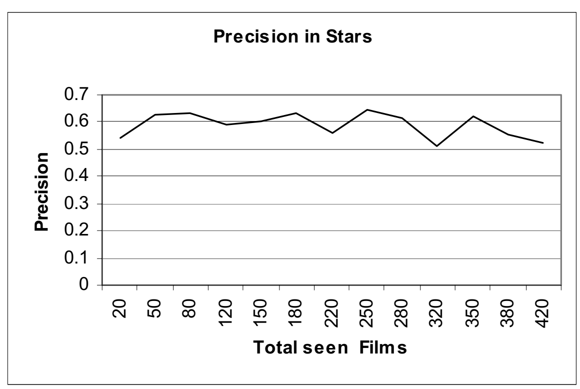
By applying the described rule sequence, we obtained:

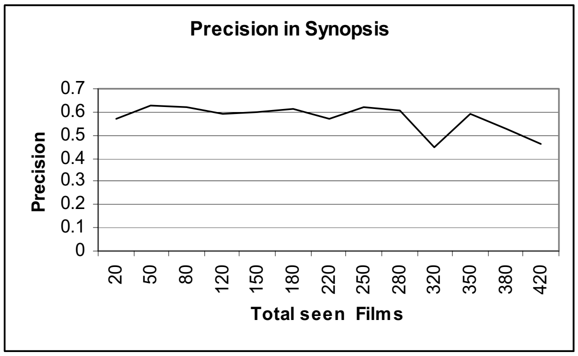
Overall percentage of successful recommendations	82%
Mean success rate per user	80%
Users with null success rate	21
Percentage of users with zero successes	2.23%

It can be observed that we are unable to make recommendations to only 2.23% of users, while the mean success rate per user is 80%, practically meaning that 4 out of 5 films are successfully recommended. This percentage is particularly high, if compared with the precision of other recommender systems.

The overall system performance could be enhanced by improving the contribution of each individual criterion. The following figures depict precision as a function of the total number of films seen, based on each of the three content criteria.







Although it seems rather strange that the precision falls as the user has evaluated many movies, it can be easily explained. As the user keeps evaluating movies, it is possible that he/she has covered a wide range of films that share a characteristic feature (Kinds, Stars, Synopsis), while being totally different and, subsequently, differently evaluated. Therefore, an interesting issue for further work would be to find a way so as to distinguish the best examples that increase performance.

Another area of research is to enrich the set of films a user has seen, when this set is too small. As can be seen from the above graphs, the performance is below average when the total number of seen films is small. We are currently working on the following idea: initially, the films are clustered using a simple K-means algorithm, based on the specific feature (Kinds, Stars, Synopsis). The mean grade of the evaluated movies that belong to each cluster is calculated and this value is assigned to the entire cluster. Then, the most characteristic film of a big enough cluster is found, compared to the ones a user has seen, and added to the training set. The first results seem promising, but further experimentation is necessary.

6. References

[1] M. Montaner, "*A Taxonomy of Recommender Agents on the Internet",* Artificial Intelligence Review 19, 2003, pp. 285–330.

[2] N. Good, J. Schafer, J. Konstan, A. Borchers, B. Sarwar, J. Herlocker and J. Riedl, "*Combining Collaborative Filtering with Personal Agents for Better Recommendations",* In Proceedings of AAAI, Vol. 35, 1999, pp. 439–446.

[3] L. Qing and M.K. Byeong, "*An Approach for Combining Content-based and Collaborative Filters"*, In Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages, 2003, pp. 17-24.

[4] M.J. Pazzani, "*A Framework for Collaborative, Content-Based and Demographic Filtering"*, Artificial Intelligence Review 13, 1999, pp. 393-408.

[5] S. Grant and G. McCalla, "*A hybrid approach to making recommendations and its application to the movie domain"*. In 2001 Canadian AI Conference, 2001, pp. 257-266

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

[7] M. Rajatish, S.D. Partha, J. Gerdur and S. Sandip, "*Movies2Go - an online voting based movie recommender system"*, Fifth International Conference on Autonomous Agents, Montreal, Canada, 2001.

[8] http://movielens.umn.edu

[9] http://www.internetworldstats.com/stats.htm

[10] F. Sebastiani, "*A tutorial on automated text categorisation"*, In Proceedings of ASAI-99, 1st Argentinian Symposium on Artificial Intelligence, 1999, pp. 7-35.

[11] C. Lueg and C. Landolt, "*A Java-Based Approach to Active Collaborative Filtering"*. Summary of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 98), Los Angeles, USA, April 18.-23., 1998, pp. 319-320.

[12] A.M.A. Wasfi, "*Collecting User Access Patterns for Building user Profiles and Collaborative Filtering"*, In Int. Conference on Intelligent User Interfaces, 1999, pp.57- 64.

[13] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm and J. Riedl, "*GroupLens: An open architecture for collaborative filtering of Netnews"*, In Proceedings ACM Conf. on Computer-Supported Cooperative Work, 1994, pp. 175-186.

[14] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "*Item-based Collaborative Filtering Recommendation Algorithms"*, In Proceedings Tenth Int. WWW Conf., 2001, pp. 285-295.

[15] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, "*An algorithmic framework for performing Collaborative Filtering"*, In Proceedings ACM-SIGIR Conference, 1999, pp. 230-237.