Multimed Tools Appl (2008) 36:55–70
DOI 10.1007/s11042-006-0082-7

A hybrid approach for movie recommendation

George Lekakos · Petros Caravelas

Published online: 21 December 2006

Abstract Collaborative and content-based filtering are the major methods in recommender systems that predict new items that users would find interesting. Each method has advantages and shortcomings of its own and is best applied in specific situations. Hybrid approaches use elements of both methods to improve performance and overcome shortcomings. In this paper, we propose a hybrid approach based on content-based and collaborative filtering, implemented in MoRe, a movie recommendation system. We also provide empirical comparison of the hybrid approach to the base methods of collaborative and content-based filtering and draw useful conclusions upon their performance.

Keywords Recommender systems · Collaborative filtering · Content-based filtering · Hybrid methods

1 Introduction

The vast amount of available information over electronic platforms (such as the Internet) urged the development of systems that filter out irrelevant information and select content that meet user needs. Recommender systems emerged in the mid-90s in order to facilitate the above process, and can be described as systems "*that produce individualized recommendations as output or have the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options*" [6]. They have proven successful in domains such as books [15], TV program [10, 16, 18, 27], jokes [9], news articles [22].

The recent advances in digital television and set-top technology with increased storage and processing capabilities enable the application of recommendation technologies in the

G. Lekakos (✉) · P. Caravelas
ELTRUN-the eBusiness center, Department of Management Science and Technology,
Athens University of Economics and Business, 47 Evelpidon Str., 11362 Athens, Greece
e-mail: glekakos@aueb.gr

P. Caravelas
e-mail: pcaravel@aueb.gr

television domain. For example products currently promoted through broadcasted advertisements to unknown recipients may be recommended to specific viewers who are most likely to respond positively to these messages. In this way recommendation technologies provide unprecedented opportunities to marketers and suppliers to target their products more effectively while reducing viewers' advertising clutter caused by the large amount of irrelevant messages [14]. Moreover, the large number of available digital television channels increases the effort required to locate content (such as movies) that it is most likely to match viewer's interests. Movie recommendation applications typically realized through the Electronic Program Guide (EPG) has been one of the first and most important research directions in the digital television domain [25].

Recommendation methods attempt to make predictions concerning users' interest on unobserved items based on previously evaluated items and/or item features. The most popular recommendation methods are collaborative filtering and content-based filtering. The first method is based on the assumption that users who have agreed on their previous evaluations would eventually agree in the future [22]. In contrast, content-based filtering considers the previous evaluations of the user that prediction refers to (called the active user) and upon them it predicts the future preferences. Both methods present advantages and disadvantages and significant research effort has been devoted to hybrid recommendation methods that combine collaborative and content-based filtering exploiting the advantages of both methods (e.g. [4, 6, 25]).

In this paper we propose a hybrid approach that combines content-based and collaborative filtering based on the monitoring of certain parameters that trigger either a content-based or a collaborative filtering prediction. We implement the hybrid, the collaborative, and the content-based filtering algorithms in a movie recommendation system called MoRe (Movie Recommender). The MoRe's performance is empirically evaluated by measuring its predictive accuracy of the algorithms as well as other important indicators such as the percentage of items that the system can actually make predictions (called prediction coverage), and the time required for run-time predictions. The remaining of the paper is organized as follows: in the next section the main recommendation techniques are described along with their advantages and limitations. In section three, the MoRe system overview is presented and in section four the algorithms implemented are described in detail. In section five, the empirical evaluation results are discussed and in the final section of the paper conclusions and future research are presented.

2 Background theory

2.1 Collaborative and content-based filtering

The typical recommendation process takes as input a user's evaluation on observed items. This evaluation is usually expressed in the form of a rating collected either implicitly (e.g., system monitors browsing behavior) or explicitly when users are asked to provide their ratings e.g., in a one-to-five scale. Ratings are then used by a recommendation approach in combination with other users' ratings and/or item features in order to produce recommendations that match the user's interests.

To produce recommendations using collaborative filtering, the active user's (the user that prediction refers to) similarities with the remaining of the users are calculated using a correlation measure (typically Pearson correlation coefficient). Then, the group (neighborhood) of the users that are the most similar to the active user is selected and their ratings are combined to produce predictions. Ratings predictions may typically lead to the presentation of a ranked or a top-n list of the most relevant (to the active user) items.

In order to produce content-based recommendations, items have to be described by some features. For example, in the book recommendation domain, the author, the genre and the most frequently used words could serve as features. Metrics such as TF-IDF and Information Gain (IG) are commonly used to extract these features [2, 20]. The items that the active user has rated are used to create a user profile. All the unrated items are compared with this profile and the most similar ones are presented to the active user.

Content-based recommendation methods can be applied in domains where it is possible to describe item by features. In domains such as movies, videos, or music, features extraction is hard to achieve [3]. Moreover qualitative criteria, such as the item's quality may be more important than content features. In contrast collaborative filtering is based on the subjective "overall taste" criterion expressed through the rating and therefore is applicable to any type of content without requiring its analysis into features. Moreover, content-based recommendations often become overspecialized, since the system tends to recommends only items within the spectrum of the previously liked items. On the contrary, collaborative filtering may provide unexpected recommendations concerning items that the user may have not considered. However, collaborative filtering has shortcomings too, the most important of which is data sparsity. Users typically rate a small fraction of the item set and therefore it is often difficult to find users with a sufficient number of overlapping ratings upon which the similarity between users is computed. This leads to low-accuracy predictions or even to failure to make predictions. In addition when a new user (new user problem) enters the system then no predictions can be made (no ratings have been provided) similarly to the situation where a new item is added in the system and no user has rated this item. The new item problem does not affect content-based recommendations, since predictions are made solely on the basis of the active user's ratings.

2.2 Hybrid approaches

In order to exploit the advantages of the recommendation methods above several hybrid approaches have been proposed, in their vast majority concerning combinations of content-based and collaborative filtering [2, 7, 23, 25] or extension the two methods by demographics-based predictions [19], while few of them utilize knowledge-based techniques where domain functional knowledge is exploited (e.g. [6]). A significant part of research in hybrid recommender systems concerns the techniques that can be used to combine the approaches since they may significantly affect the prediction outcome.

Burke [6] classifies hybridization techniques into seven classes: *weighted* where each of the recommendation approaches makes predictions which are then combined into a single prediction; *switching* where one of the recommendation techniques is selected to make the prediction when certain criteria are met; *mixed* in which predictions from each of the recommendation techniques are presented to the user; *feature combination* where a single prediction algorithm is provided with features from different recommendation techniques; *cascade* where the output from one recommendation technique is refined by another; *feature augmentation* where the output from one recommendation technique is fed to another, and *meta-level* in which the entire model produced by one recommendation technique is utilized by another.

Switching, mixed, and weighted hybrids are differentiated from the remaining techniques in Burke's taxonomy by the fact that each of the individual (base) recommendation methods produce independently from each other a prediction which is then presented to the user either as a single combined prediction (switching, weighted) or as two independent predictions (mixed). Switching hybrids in particular, are low-complexity hybridization methods based on the examination of the conditions that affect the performance of the base algorithms each time a prediction is requested. When certain conditions occur the final prediction is the outcome of the base recommendation approach that is not affected (or is less affected) from these conditions. For example Billsus and Pazzani [5], in DailyLearner, a system for the provision of personalized news stories, propose an approach which is based upon two types of user models: a short-term and a long-term user model referring to the modeling of user's short and long-term interests. The switching hybrid strategy applies when a story cannot be classified via the short-term model due to the absence of similar stories above a fixed threshold. The classification task is then passed to the long-term model, which contains the TF-IDF weights that characterize news categories and executed by applying the Naïve Bayes model. Tran and Cohen [26] combine collaborative and knowledge-based recommendation approaches for product recommendations. The knowledge-based approach is implemented by collecting information (through dialogue) concerning product features and related importance value and subsequently exploiting domain knowledge to match products to the user needs. Although the knowledge-based process is resource intensive requiring human knowledge-engineering effort, it can provide recommendations even in the complete absence of user ratings. The system monitors the number of users with known interest profiles and the number of rated items in the database. If either of the two variables is below a fixed threshold, then the knowledge-based recommendation is presented to the user, otherwise the collaborative approach is applied.

In collaborative filtering several conditions may affect its performance such as the sparsity of ratings discussed above. In the absence of sufficient amount of available ratings for the active user similarities may be lead to erroneous selection of actually "bad" neighbors as good ones and vice versa. The size of the neighborhood also affects prediction accuracy. If too few neighbors contribute in the final prediction for the active user then this leads to reduced accuracy [12]. The size of the neighborhood depends on the neighbor selection strategy. In the top-n strategy the size of the neighborhood (n) is predetermined but the quality of the neighbors as recommenders still depends on their level of similarity to the active user. If users with low correlation to the active user are included in his/her neighborhood the prediction accuracy is negatively affected. In the threshold-based strategy setting a high similarity threshold value may lead to more accurate prediction but the final neighborhood size (that also affect accuracy) depends on the number of users with correlation above the selected threshold. Moreover, there is trade-off between accuracy and coverage with respect to the size of the neighborhood. If only the most similar users contribute to the final prediction then prediction accuracy is improved but at the risk of tracing only a limited number of such neighbors, which in turn may result in reduction in coverage. Herlocker et al. [12] suggest that the optimum results for both accuracy and coverage are achieved for a zero threshold value.

In content-based filtering the features used to describe the content are of primary importance. The more descriptive they are the more accurate the prediction is, provided that the active user has rated a sufficient number of items similar to the target item. In computational terms, content-based prediction can be performed even if the users has rated at least one similar item though prediction accuracy increases with the number of similar items. Content-based filtering is not affected by the number of users who have rated the target item since it operates solely upon the active user's ratings.

It has been empirically shown that collaborative filtering is more accurate than content-based filtering provided that certain criteria are met [1, 4]. As discussed above two criteria with significant effect are the neighborhood size and the number of rated items by the active user. In this paper we present a switching hybrid algorithm whose main prediction approach is based on collaborative filtering switched to content-based filtering when the above criteria are met. Typically, hybrid recommendation methods aim at providing more accurate predictive results compared to the base methods. The proposed hybrid approach besides predictive accuracy also considers two other factors with practical significance: the prediction coverage as well as the time required to make a (run-time) prediction. The MoRe system where the hybrid and the base algorithms (content-based and collaborative filtering) have been implemented, serves as the tool for the empirical evaluation of the algorithms.

3 MoRe system overview

The MoRe system is a Web-based recommender system that collects user ratings concerning movies on one-to-five scale through its graphical user interface. More specifically as soon as a new user is registered with the system he/she is asked to provide a number of ratings in order for the system to initiate the prediction process (new user problem). The selection of movies that are presented to the user is based on a measure proposed by Rashid et al. [21] computed as log(popularity)*entropy. The selection of the most popular movies increases the possibility to collect the respective ratings since it is most likely that these movies have been actually seen by the new user. The collected ratings are organized in a user × item matrix and combination with the movies dataset are loaded into the recommendation algorithms, which have been implemented in MoRe (Fig. 1): a pure collaborative filtering, a content-based, and the proposed hybrid approach that has been implemented in two versions (called switching and substitute). The two versions of the hybrid algorithm are differentiated by the parameter that controls the switch from collaborative filtering to content-based filtering as will be analyzed in the following sections.

The MoRe system utilizes a version of the well-known MovieLens dataset http://www.movielens.org) that contains a million user ratings provided by 6,040 original MovieLens users for about 4,000 movies. Each user has rated at least 20 movies in the one-to-five rating scale and the sparsity of the user ratings matrix is 95.8%. Since the dataset contains only the name and the production year of each movie, it is necessary to augment the movie description features for the content-based predictor. To accomplish that, we implemented a web crawler that seeks for data in the website of the Internet Movie Database (IMDb http://www.imdb.com). The crawler exploits the search tool of IMDb and collects data about the genre, cast, director, writing credits, producers and plot keywords of each movie. The number of keywords may exponentially increase the number of features used to describe the features and therefore the system administrator may remove keywords from the movie description that appear in less than a certain number of movies. In addition, the system creates (at an off-line phase) the set of most similar movies for all available movies in order to speed-up run-time predictions. The size of the set of most similar movies is determined by the system administrator (Fig. 2).

In order to make recommendations, collaborative filtering uses the ratings matrix while the content-based predictor uses mainly the movie data files. Hybrid methods use both the content-based and the collaborative engines. The system is able to produce recommendations with more than one method, but at any given time only one method is applied determined by administrator of the system (Fig. 3).

Fig. 1 More system overall design

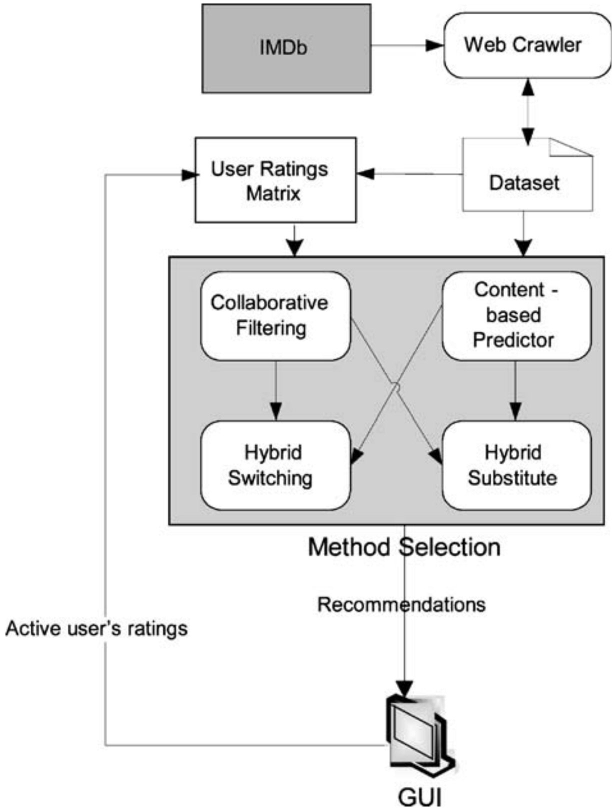

Fig. 2 Selection of features and pre-computation of similar movies sets

Fig. 3 Method selection in MoRe

Fig. 4 Ranked list of movie recommendations

Users receive the recommendations in a ranked list of movies where the prediction appears to the user in a “five-star” scale, while users may provide their feedback directly on the recommended movies (Fig. 4).

The system can also be used for experimental purposes. The system administrator may select the size of the training and test sets as percentage of the whole dataset and initiate the estimation of the accuracy of the recommendation methods in MoRe, which are analyzed in the next sections.

4 Recommendation algorithms

4.1 Pure collaborative filtering

Our collaborative filtering engine applies the typical neighbourhood-based algorithm [22], divided into three steps: (a) computation of similarities between the active and the remaining of the users, (b) neighborhood development, and (c) computation of prediction based on weighted average of the neighbors’ ratings on the target item.

For the first step, typically the Pearson correlation coefficient is used (formula 1).

$$r = \frac{\sum_i \left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sqrt{\sum_i \left(X_i - \overline{X}\right)^2 \sum_i \left(Y_i - \overline{Y}\right)^2}} \tag{1}$$

where X_i and Y_i are the ratings of users X and Y for movie i, and $\overline{X}$, $\overline{Y}$ refer to the mean values of the available ratings for the users X and Y. However, in the MoRe implementation we used formula 2 below which is equivalent to formula 1 but it computes similarities faster since it does not need to compute the mean rating values (n represents the number of commonly rated movies by users X and Y):

$$r = \frac{n\sum_i X_i Y_i - \sum_i X_i \sum_i Y_i}{\sqrt{n\sum_i X_i^2 - \left(\sum_i X_i\right)^2}\sqrt{n\sum_i Y_i^2 - \left(\sum_i Y_i\right)^2}} \tag{2}$$

Note that in the above formulas if either user has evaluated all movies with identical ratings the result is a “divide by zero” error and therefore we decided to ignore users with such ratings. In addition we devaluate the contribution of neighbors with less than 50 commonly rated movies by applying a significance weight of $n/50$, where n is the number of ratings in common [11].

At the neighborhood development step of the collaborative filtering process we select neighbors with positive correlation to the active user. In order to increase the accuracy of the recommendations, prediction for a movie is produced only if the neighbourhood consists of at least five neighbors.

To compute an arithmetic prediction for a movie, the weighted average of all neighbors’ ratings is computed using the following formula 3:

$$K_i = \overline{K} + \frac{\sum_{J \in \text{Neighboors}} \left(J_i - \overline{J}\right) r_{\text{KJ}}}{\sum_J \left|r_{\text{KJ}}\right|}, \tag{3}$$

where K_i is the prediction for movie i, $\overline{K}$ is the average mean of active user's ratings, J_i is the rating of neighbour J for the movie i, $\overline{J}$ is the average mean of neighbour J's ratings and r_{KJ} is the Pearson correlation measure for active user and neighbor J.

4.2 Pure content-based predictor

In the content-based prediction we consider as features all movie contributors (cast, directors, writers, and producers), the genre, and the plot words. Features that appear in only one movie are ignored. Each movie is represented by a vector, the length of which is equal to the number of non-unique features of all available movies. The elements of the vector state the existence or non-existence (Boolean) of a specific feature in the description of the movie.

To calculate the similarity of two movies, we use the cosine similarity measure computed as:

$$\cos\left(\overrightarrow{\boldsymbol{a}}, \overrightarrow{\boldsymbol{b}}\right) = \frac{\overrightarrow{\boldsymbol{a}} \cdot \overrightarrow{\boldsymbol{b}}}{\left\|\overrightarrow{\boldsymbol{a}}\right\| * \left\|\overrightarrow{\boldsymbol{b}}\right\|} = \frac{\sum_i a_i b_i}{\sqrt{\sum_i a_i^2}\sqrt{\sum_i b_i^2}} \tag{4}$$

In the formula above, a_i and b_i are the values of the ith elements of vectors $\overrightarrow{a}$ and $\overrightarrow{b}$.

The algorithm we use to produce recommendations is an extension of the top-N item-based algorithm that is described in [13]. Since the movie set does not change dynamically when the system is online, the similarities between all pairs of movies in the dataset are pre-computed off-line and for each movie the k-most similar movies are recorded, along with their corresponding similarity values. When a user that has rated positively (i.e four or five) a set U of movies, asks for recommendations, a set C of candidate movies for recommendation is created as the union of the k-most similar movies for each movie j ∈ U, excluding movies already in U. The next step is to compute the similarity of each movie c ∈ C to the set U as the sum of the similarities between c ∈ C and all movies j ∈ U. Finally, the movies in C are sorted with respect to that similarity (Fig. 5). Note that typically content-based recommendation is based upon the similarities between items' features and user's profile consisting of preferences on items' features. Instead, Karypis computes similarities between items upon all users' ratings completely ignoring item features (this approach is also known as item-to-item correlation). However, the above approach (that can also be regarded as content-based retrieval), we extend Karypis' algorithm by utilizing the movies' features rather that the users' ratings to find the most similar movies to the ones that the user has rated positively in the past and therefore we preserve the term content-based filtering.

Since we are interested in numerical ratings in order to combine content-based and collaborative filtering predictions, we extend Karypis' [13] algorithm (which is designed for binary ratings) as follows: let MaxSim, MinSim is the maximum and minimum similarities for each movie in $c \in C$ to U, and Sim_i the similarity of a movie M_i to the set U. The numerical prediction Pr_i for the movie is computed as:

$$\mathrm{Pr}_i = \frac{(\mathrm{Sim}_i - \mathrm{MinSim})*4}{(\mathrm{MaxSim} - \mathrm{MinSim})} + 1 \tag{5}$$

Formula 5 above normalizes similarities from [MaxSim, MinSim] to [1,5] that is the rating scale used in collaborative filtering. For example, if $\mathrm{Sim_i}$=0.8, MinSim=0.1, MaxSim=0,9 then $\mathrm{Pr_i}$=4.5. Note that the formula applies for any similarity value (above or below one).

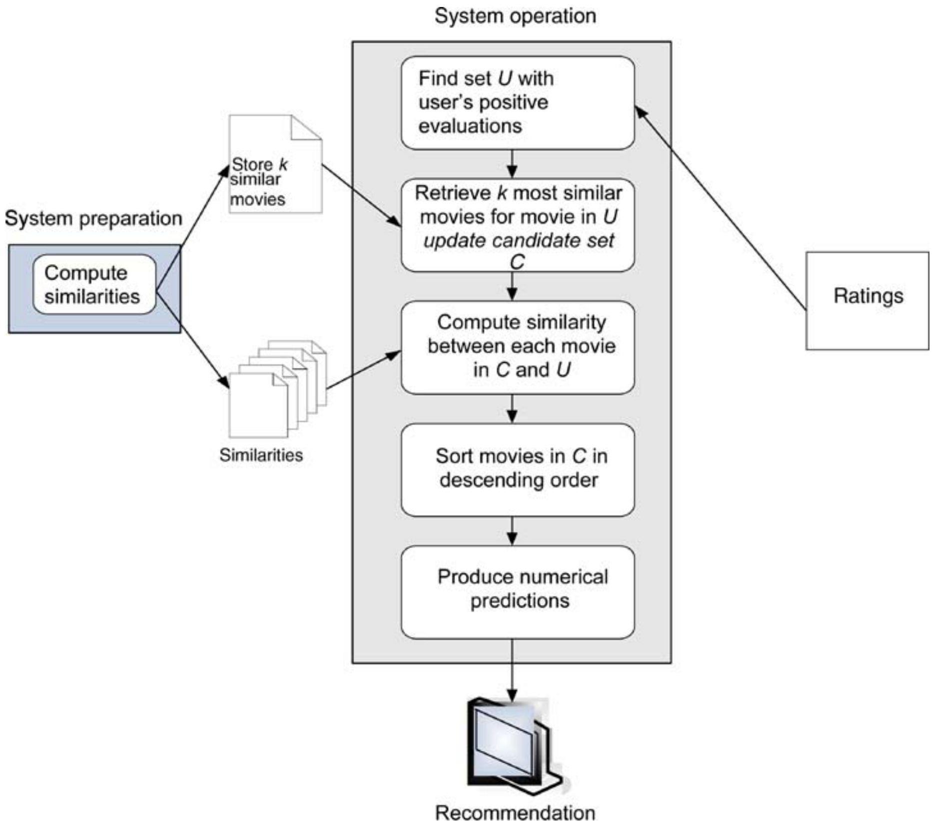

Fig. 5 Content-based filtering prediction process

Due to the fact that movie similarities are computed offline, we are able to produce content-based recommendations much faster than collaborative filtering recommendations. Moreover, in contrast to collaborative filtering, content-based predictions can always be produced for the specific dataset.

In addition, we implemented content-based filtering using the Naïve Bayes algorithm. Each of the five numerical ratings is considered as a class label and prediction u for an item is computed using formula 6:

$$u = \underset{u_j \in \{1,2,3,4,5\}}{\arg\max} P(u_j) \prod_{i=0}^{m} P(a_i | u_j) \tag{6}$$

where u_j is the rating provided by the user (u_j=1,2,3,4,5), $P(u_j)$ is the probability that any item can be rated by the user with u_j (computed by the available user ratings), m is the number of terms used in the description of the items and $P(a_i|u_j)$ is the probability to find in the item's description the term a_i when it has been rated with u_j. The probability $P(a_i|u_j)$ is computed by formula 7:

$$P(a_i|u_j) = \frac{n_i + 1}{n + |\text{Vocabulary}|} \tag{7}$$

where n is the total number of occurrences of all terms that are used for the description of the items and have been rated with u_j, n_i is the frequency of appearance of the term a_i in the n terms, and |Vocabulary| is the number of unique terms appearing in all items that have been rated by the user. The Naïve Bayes algorithm has been successfully used in the book recommendation domain by Mooney and Roy [17].

4.3 Hybrid recommendation methods

The proposed hybrid recommendation method is implemented in two variations. The first one called *substitute* aims to utilize collaborative filtering as the main prediction method and switch to content-based when collaborative filtering predictions cannot be made. The use of collaborative filtering as the primary method is based on the superiority of collaborative filtering in the movie domain and empirical experimentation. Content-based predictions are triggered when the neighborhood size of the active user consists of less than five users. This approach is expected to increase both the prediction accuracy as well as the prediction coverage. Indeed, the collaborative filtering algorithm described above requires at least five neighbors for an active user in order to make a prediction. This requirement increases the accuracy of the collaborative filtering method itself (compared to the typical collaborative filtering algorithm) but leads to a prediction failure when it is not met. For these items (for which prediction cannot be made) content-based prediction is always feasible and therefore the overall accuracy of the substitute hybrid algorithm is expected to improve compared to the collaborative filtering as well as the content-based algorithms. Although this approach is also expected to improve prediction coverage, the time required to make predictions may increase due to the additional steps required by the content-based algorithm. However, this delay may be practically insignificant since the time needed to make content-based recommendations is significantly shorter than the time to produce recommendations with collaborative filtering.

The second variation of the proposed hybrid approach called *switching* is based on the number of available ratings for the active user as the switching criterion. Collaborative filtering prediction is negatively affected when few ratings are available for the active user. In contrast, content-based method deal with this problem more effectively since predictions can be produced even upon few ratings. The switching hybrid uses collaborative filtering as the main recommendation method and triggers a content-based prediction when the number of available ratings falls below a fixed threshold. This threshold value can be experimentally determined and for the specific dataset has been set to 40 ratings. In terms of prediction coverage the switching hybrid is not expected to differ significantly from the collaborative filtering since content-based coverage may be applied even if a collaborative filtering prediction can be produced, in contrast to the substitute hybrid which triggers content-based prediction upon the “failure” of collaborative filtering to make predictions. Although the two variations above follow the exactly the same approach (collaborative filtering as the main method) they differ in the switching criterion with distinct implementations in the MoRe system and therefore discussed separately below.

5 Experimental evaluation

The objective of the experimental evaluation is to compare the two versions of the hybrid algorithm against each other as well as against the base algorithms (collaborative and content-based filtering). The comparison is performed in terms of predictive accuracy, coverage, and actual time required for run-time predictions. Moreover, since the "pure collaborative filtering" implemented in MoRe adopts a neighborhood-size threshold (five neighbors), we will examine its performance against the typical collaborative filtering method without the neighborhood size restriction. We will also also evaluate the effect of the number of features used to describe the movies on the prediction accuracy of the content-based algorithm.

The evaluation measures utilized for estimating prediction accuracy is the Mean Absolute Error (MAE). The Mean Absolute Error [24] is a suitable measure of precision for systems that use numerical user ratings and numerical predictions. If $r_1,...,r_n$ are the actual ratings of a user in the test set, $p_1,...,p_n$ are the predicted ratings, and $E = \varepsilon_1, \ldots, \varepsilon_n = \{p_1 - r_1, \ldots, p_n - r_n\}$ are the errors, then the Mean Absolute Error is computed as:

$$\mathrm{MAE} = \left|\overline{E}\right| = \frac{\sum_{i=1}^{n} |\varepsilon_l|}{n}$$

In the experimental process the original dataset is separated in two subsets randomly selected: a training set containing the 80% of ratings of each available user, and a test set including the remaining 20% of the ratings. The ratings that belong to the test set are ignored by the system and we try to produce predictions for them using only the remaining ratings of the training set. To compare the MAE values of the different recommendation methods and to verify that the differences are statistically significant we apply non-parametric Wilcoxon rank test, in the 99% confidence space (since normality requirement or parametric test is not met).

The MAE for the pure collaborative filtering method is 0.7597 and the coverage 98.34%. The MAE value for collaborative filtering method (without the neighborhood size restriction) is 0.7654 and the respective coverage 99.2%. The p-value of the Wilcoxon test (p=0.0002) indicates a statistically significant difference suggesting that the restriction to produce prediction for a movie only if the neighbourhood consists of at least five neighbours lead to more accurate predictions, but scarifies a portion of coverage.

The pure content-based predictor presents MAE value 0.9253, which is significantly different (p=0.000) than collaborative filtering. The coverage is 100%, since content-based predictions ensures that prediction can always be produced for every movie (provided that the active user has rated at least one movie). In the above experiment we used a word as a feature if it appeared in the description of at least two movies. We calculated the accuracy of the predictions when this threshold value is increased to three, five, ten and fifteen movies, as shown in Table 1.

Comparing cases one and two above we found no significant differences while all difference between 2 and 3, 4, 5 (p=0.0000 for all cases) cases are statistically significant.

Thus, we may conclude that the number of features that are used to represent the movies is an important factor of the accuracy of the recommendations and, more specifically, the more features are used, the more accurate the recommendations are. Note that the Naïve Bayes

Table 1 Number of features and prediction accuracy

Case	Threshold (movies)	MAE	Number of features
1	2	0.9253	10,626
2	3	0.9253	10,620
3	5	0.9275	7,865
4	10	0.9555	5,430
5	15	0.9780	3,514

algorithm performed poorly in terms of accuracy with MAE=1.2434. We improved its performance when considered ratings above three as positive ratings and below three as negative (MAE=1.118). However, this error is still significantly higher than the previous implementation and therefore we exclude it from the development of the hybrid approaches.

Substitute Hybrid Recommendation Method was designed to achieve 100% coverage. The MAE of the method was calculated to be 0.7501, which is a statistically important improvement of the accuracy of "pure" collaborative filtering ($p<0.00001$).

The coverage of the Switching Hybrid Recommendation Method is 98.8%, while the MAE was 0.7702, which is a statistically different in relevance to substitute hybrid and "pure" collaborative filtering methods ($p=0.000$). This method produces recommendations of less accuracy than both "pure" collaborative filtering and substitute hybrid has greater coverage than the first and lower that the latter method, but it produces recommendations in reduced time than both methods above. Even though recommendation methods are usually evaluated in terms of accuracy and coverage, the reduction of execution time might be considered more important for a recommender system designer, in particular in a system with a large number of users and/or items.

Table 2 depicts the MAE values, coverage, and time required for run-time prediction (on a Pentium machine running at 3.2 GHz with 1 GB RAM) for all four recommendation methods.

Note that the most demanding algorithm in terms of resources for run-time prediction is collaborative filtering. If similarities are computed between the active and the remaining users at prediction time then its complexity is O(nm) for n users and m items. This may reduced at O(m) if similarities for all pairs or users are pre-computed with an off-line cost O(n^2m). However, such a pre-computation step affects one of the most important characteristics of collaborative filtering, which is its ability to incorporate the most up-to-date ratings in the prediction process. In domains where rapid changes in user interests are not likely to occur the off-line computation step may be a worthwhile alternative.

6 Conclusions and future research

The above empirical results provide useful insights concerning collaborative filtering and content-based predictions as well as its combination under the substitute and switching hybridization mechanisms.

Collaborative filtering remains one of the most accurate recommendation methods but for very large datasets the scalability problem may be significant and a similarities pre-computation phase may reduce the run-time prediction cost. The size of active user's neighbourhood does affect the accuracy of recommendations. Setting the minimum number of neighbors to five improves prediction accuracy but at a small cost in coverage.

Table 2 MAE, coverage, and prediction time for the recommendation methods

	MAE	Coverage	Run time prediction
Pure collaborative filtering	0.7597	98.34%	14 s
Pure content-based recommendations	0.9253	100%	3 s
Substitute hybrid recommendation method	0.7501	100%	16 s
Switching hybrid recommendation method	0.7702	98.8%	10 s

Content-based recommendations are significantly less accurate than collaborative filtering, but are produced much faster. In the movie recommendation domain, the accuracy depends on the number of features that are used to describe the movies. The more features are used, the more accurate recommendations are.

Substitute hybrid recommendation method improves the performance of collaborative filtering in terms of both accuracy and coverage. Although the difference in coverage of collaborative filtering on the specific dataset and with specific conditions (user rated at least 20 movies, 0 weight threshold value) is rather insignificant, it has been reported that this is not always the case, in particular when increasing the weight threshold value [11]. On the other hand, the switching hybrid recommendation method fails to improve the accuracy of collaborative filtering, but significantly reduces execution time.

The MoRe system is specifically designed for movie recommendations but its collaborative filtering engine may be used for any type of content. The evaluation of the algorithms implemented in the MoRe system was based on a specific dataset which limits the above conclusions in the movie domain. Thus, it would be useful to evaluate the system on alternative datasets in other domains as well in order to examine the generalization ability of our conclusions.

As future research it would also be particularly valuable to perform an experimental evaluation of the system, as well as the proposed recommendations methods, by actual users. This would allow for checking whether the small but statistically significant differences on recommendation accuracy are detectable by the users. Moreover, it would be useful to know which performance factor (accuracy, coverage or execution time) is considered to be the most important by the users, since that kind of knowledge could set the priorities of our future research.

Another issue that should be subject for future research is the way of the recommendations presented to the users, the layout of the graphical user interface and how this influences the user ratings. Although there are some studies on these issues (e.g., [8]), it is a fact that the focus in recommender system research is on the algorithms used in the recommendation process.

References

1. Alspector J, Koicz A, Karunanithi N (1997) Feature-based and clique-based user models for movie selection: a comparative study. User Model User Adapt Interact 7:297–304
2. Balabanovic M (1997) An adaptive web page recommendation service. In Proceedings of the ACM first international conference on autonomous agents. Marina del Ray, California, pp 378–385
3. Balabanovic M, Shoham Y (1997) Fab: content-based collaborative recommendation. Commun ACM 40 (3):66–72
4. Basu C, Hirsh H, Cohen W (1998) Recommendation as classification: using social and content-based information in recommendation. In Proceedings of the 15th national conference on artificial intelligence. Madison, WI, pp 714–720
5. Billsus D, Pazzani M (2000) User modeling for adaptive news access. User Model User Adapt Interact 10(2–3):147–180
6. Burke R (2002) Hybrid recommender systems: survey and experiments. User Model User Adapt Interact 12:331–370
7. Claypool M, Gokhale A, Miranda T, Murnikov P, Netes D, Sartin M (1999) Combining content-based and collaborative filters in an online newspaper. In Proceedings of the ACM SIGIR workshop on recommender systems. Berkeley, CA, http://www.csee.umbc.edu/~ian/sigir99-rec/
8. Cosle D, Lam S, Albert IK, Riedl J (2003) Is seeing believing? How recommender systems influence users' opinions. In Proceedings of the computer–human interaction 2003 conference on human factors in computing systems. Fort Lauderdale, FL, pp 585–592

9. Goldberg K, Roeder T, Guptra D, Perkins C (2001) Eigentaste: a constant-time collaborative filtering algorithm. Inf Retr 4(2):133–151
10. Gutta S, Kuparati K, Lee K, Martino J, Schaffer D, Zimmerman J (2000) TV content recommender system. In Proceedings of the 17th national conference on artificial intelligence. Austin, Texas, pp 1121–1122
11. Herlocker J, Konstan J, Borchers A, Riedl J (1999) An algorithmic framework for performing collaborative filtering. In Proceedings of the twenty-second international conference on research and development in information retrieval (SIGIR '99). New York, pp 230–237
12. Herlocker J, Konstan J, Riedl J (2002). An empirical analysis of design choices in neighborhood-base collaborative filtering algorithms. Inf Retr 5:287–310
13. Karypis G (2001) Evaluation of item-based top-n recommendation algorithms. In Proceedings of the CIKM 2001, pp 247–254
14. Lekakos G, Giaglis GM (2004) A lifestyle-based approach for delivering personalized advertisements in digital interactive television. J Comput Mediat Commun 9(2)
15. Linden G, Smith B, York J (2003) Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Computing 2003, pp 76–80, January–February
16. Miller BN, Albert I, Lam SK, Konstan J, Riedl J (2003) MovieLens unplugged: experiences with an occasionally connected recommender system. In Proceedings of the international conference on intelligent user interfaces
17. Mooney RJ, Roy L (2000) Content-based book recommending using learning for text categorization. In Proceedings of the fifth ACM conference in digital libraries. San Antonio, TX, pp 195–204
18. O'Sullivan D, Smyth B, Wilson DC, McDonald K, Smeaton A (2004) Improving the quality of the personalized electronic program guide. User Model User Adapt Interact 14(1):5–36
19. Pazzani M (1999) A framework for collaborative, content-based and demographic filtering. Artif Intell Rev 13(5/6):393–408
20. Pazzani M, Billsus D (1997) Learning and revising user profiles: the identification of interesting web sites. Mach Learn 27:313–331
21. Rashid A, Albert I, Cosley D, Lam S, McNee S, Konstan J, Riedl J (2002) Getting to know you: Learning new user preferences in recommender systems. In Proceedings of the proceedings of international conference on intelligent user interfaces
22. Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J (1994) GroupLens: An open architecture for collaborative filtering of NetNews. In Proceedings of the ACM conference on computer supported cooperative work, pp 175–186
23. Schwab I, Pohl W, Koychev I (2000) Learning to recommend from positive evidence. In Proceedings of the intelligent user interfaces. New Orleans, LA, pp 241–247
24. Shardanand U, Maes P (1995) Social information filtering: algorithms for automating "word of mouth." In: Proceedings of the ACM CHI'95 conference on human factors in computing systems. Denver, Colorado, pp 210–217
25. Smyth B, Cotter P (2000) A personalized television listings service. Commun ACM 43(8):107–111
26. Tran T, Cohen R (2000) Hybrid recommender systems for electronic commerce. In Proceedings of the knowledge-based electronic markets. Austin, Texas, pp 78–84
27. Yu Z, Zhou X (2004) TV3P: An adaptive assistant for personalized TV. IEEE Trans Consum Electron 50 (1):393–399

George Lekakos Dr. George Lekakos is an adjunct Lecturer at the Department of Management Science and Technology, Athens University of Economics and Business, Athens, Greece and a Visiting Lecturer at the Department of Computer Science, University of Cyprus. He is the director of the Digital Interactive Media (DIM) research team of the ELTRUN research group within the Athens University of Economics and Business. Dr. Lekakos has worked in the area of personalized and adaptive systems, human-computer interaction, and machine learning. He has published more than thirty papers in international journals and conferences, and he is the co-editor of books and conference proceedings.

Petros Caravelas Mr. Petros Caravelas holds a B.Sc. in Computer Science from the Department of Informatics, Athens University of Economics and Business (AUEB), Athens, Greece and an MSc in Information Systems from the above University. He has been a member of the ELTRUN research group within AUEB. His research interests include recommender systems, information retrieval and data mining, artificial intelligence, genetic algorithms and machine learning.