MoviExplain: A Recommender System with Explanations

Panagiotis Symeonidis
Department of Informatics
Aristotle University
Thessaloniki, 54124
symeon@csd.auth.gr

Alexandros Nanopoulos
Institute of Computer Science
University of Hildesheim
Hildesheim, D-31141
nanopoulos@ismll.de

Yannis Manolopoulos
Department of Informatics
Aristotle University
Thessaloniki, 54124
manolopo@csd.auth.gr

ABSTRACT

Providing justification to a recommendation gives credibility to a recommender system. Some recommender systems (Amazon.com etc.) try to explain their recommendations, in an effort to regain customer acceptance and trust. But their explanations are poor, because they are based solely on rating data, ignoring the content data. Our prototype system MoviExplain is a movie recommender system that provides both accurate and justifiable recommendations.

Categories and Subject Descriptors

H.3.3 [**Information Search-Retrieval**]: Information Filtering. **General Terms:** Algorithms, Performance. **Keywords:** Recommender Systems, Explanations

1. INTRODUCTION

Recent research noticed that the acceptance of Collaborative Filtering (CF) recommender systems (like Amazon.com, MovieLens etc.) increases, when users receive justified recommendations [3]. For instance, Amazon adopted the following two styles of justification: (i) "Customers who bought item X also bought items $Y, Z, \ldots$". This is the so called "nearest neighbor" style [1] of justification. (ii) "Item Y is recommended because you rated item X". This is the so called "influence" style, where the system isolates the item, X, that influenced most the recommendation of movie Y.

Pure Content-Based filtering (CB) systems [6] make recommendations for a target user based on the past data of that user without involving data from other users. Based on pure CB, several research works [2, 6] were able to provide explanations for their recommendations. For instance, Billsus and Pazzani [2] recommend news articles to users, providing the following style of justification. "This story received a high relevance score, because it contains the words f_1, f_2, and f_3". This is the "keyword" [1] justification style.

Bilgic et al. [1] claimed that the "influence" and "keyword" styles are better than the "nearest neighbor" style, because they allow users to accurately predict their true opinion of an item. Nevertheless, both "influence" and "keyword" styles can not justify adequately their recommendations, because they are based solely either on data about ratings (rating data), or solely on *content* data, which are extracted in the form of features that are derived from the items.

Several CF systems have proposed the combination of content data with rating data [5, 7]. By combining CF with CB, data sparsity can be reduced, yielding to more accurate recommendations. For this reason, recently proposed recommender systems, like CinemaScreen [7] and Libra [1], combine CB and CF in their recommendations.

Our prototype system MoviExplain is a movie recommender system with explanations. It relies on the democratic nature of voting. In essence, MoviExplain uses a simple heuristic to interpret a rating by a user A to a movie B, as a vote to the features of movie B (actors, directors etc.). Based on these features, MoviExplain builds a feature profile for each user.

MoviExplain groups users into *biclusters*, i.e., group of users which exhibit highly correlated ratings on groups of movies, to detect partial matching of user's preferences. Each bicluster acts like a community for its corresponding movies; e.g., in a system that recommends movies, such a group may be users that prefer comedies. Moreover, by using groups instead of individual users, the extracted features are collective, reflecting preferences of whole communities. As a result, collective features cover a wider range of users preferences and result to better explanations.

The justification style of MoviExplain combines "keyword" with "influence" explanation styles [1], having the following form: "Movie X is recommended because it contains features $a, b, \ldots$ which are also included in movies $Z, W, \ldots$ you have already rated". If inside the user's feature profile, these features are frequent, this is a strong evidence for justifying the recommendations.

2. RELATED WORK

There have been several hybrid attempts to combine CB with CF. The Libra [1] System employs an approach called Content-Boosted Collaborative Filtering (CBCF) [5]. The basic idea of CBCF is to use content-based predictions to "fill out" the user-item ratings matrix. In contrast to Fab and Libra, the CinemaScreen System [7] reverses the strategy and runs firstly CF and then CB (CFCB). In particular, CinemaScreen system computes predicted rating values for movies based on CF and then applies CB to generate the recommendation list.

RecSys'09, October 23–25, 2009, New York, New York, USA.

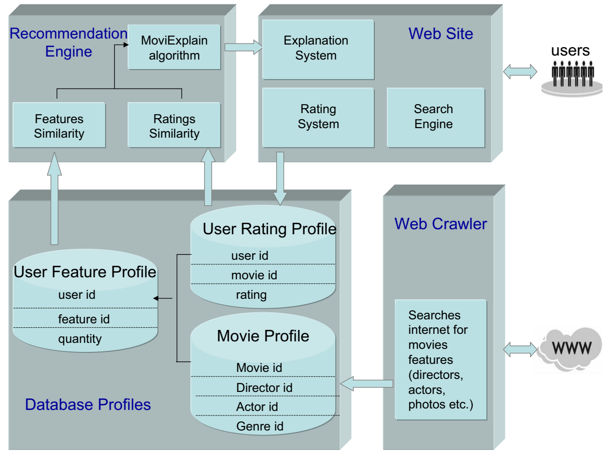


Figure 1: Components of the MoviExplain recommender system

Regarding research on explanations, many pure CB systems have tried to provide explanations to users. For instance, Billsus and Pazzani [2] recommend news articles to users, providing also explanations for reasoning their recommendations. In 2000, Mooney and Roy [6] proposed a method based also on pure CB for recommending books. These works were pioneering for the problem of explanation and inspired subsequent research on combining CF and CB for explanation purposes. In the area of CF, there is a little existing research on explaining. In 2000, Herlocker et al. [3] proposed 21 different interfaces of explaining CF recommendations. By conducting a survey, they claim that the "nearest neighbor" style is effective in supporting explanations. Amazon.com's recommender system early adopted the "nearest neighbor" explanation style. In 2005, Bilgic et al. [1] demonstrate, through a survey, that the "influence" and "keyword" styles are better than the "nearest neighbor" style, because they help users to accurately predict their true opinion of a recommendation.

3. MOVIEXPLAIN SYSTEM DESCRIPTION

MoviExplain system consists of several components. The system's architecture is illustrated in Figure 1, where the main four sub-systems are described: (i) a Web Crawler, (ii) the Database Profiles, (iii) a Recommendation Engine and (iv) the Web Site. In the following sections, we describe each sub-system of MoviExplain in details.

3.1 MoviExplain Web Crawler

MoviExplain uses a web crawler to search for information about movies on the Web. The movies information concerns the basic movies characteristics like its cast (directors and actors), their official web pages, posters and various photos, movie genres etc. Moreover, a search engine summarizes this content and adds the appropriate links to their indexes. Thus, a user can search for his favorite movie using the MoviExplain search engine and get updated information about its features. MoviExplain is fully integrated to the well-known Internet Movie Data Base (IMDB) web site.

3.2 MoviExplain Database Profiles

As described previously, MoviExplain's database profiles contain users ratings and movies' features. The feature extraction has been done from the Internet Movie Database (IMDB). In this work, following related research, e.g. [7], we select as movies' features the actors, directors, and genres.

3.3 MoviExplain Recommendation Engine

The Recommendation Engine is the heart of the MoviExplain system. It aims to provide both accurate and justifiable recommendations. The recommendation algorithm contains four stages: (i) The creation of user groups, (ii) the feature-weighting, (iii) the neighborhood formation, and (iv) the generation of the recommendation and justification lists.

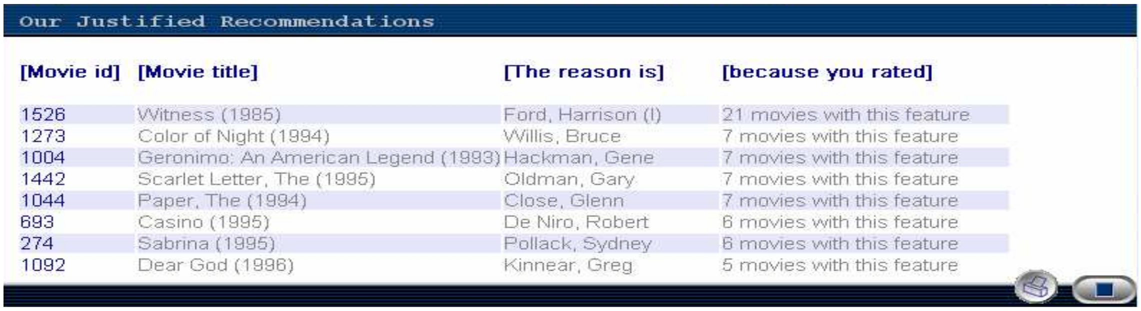
3.4 MoviExplain Web Site

Users interact with MoviExplain through its web site[1]. MoviExplain consists of 3 sub-systems: (i) the Search Engine, (ii) the Rating System and (iii) the Recommendation with Explanation System. The Search Engine keeps updated information about movies and their features, which are collected by the web-crawler. The Rating System is meant to help a user to keep track of the movies he has rated. Based on these features, MoviExplain builds a feature profile for each user. Finally, MoviExplain provides as explanation, the feature that influenced most a recommendation, showing also how strong is this feature in the feature profile of a user. As shown in Figure 2, the link "The reason is" reveals the favorite feature that influenced most the MoviExplain's recommendations, while the link "because you rated" shows how strong is this feature in the feature profile of a user.

4. EXPERIMENTAL RESULTS

In this section, we experimentally study the performance of the proposed MoviExplain System. For comparison purposes, we include as representative of the hybrid CFCB algorithms, the CinemaScreen Recommender Agent [7] denoted as CinemaScreen. As representative of the hybrid CBCF algorithms, we use the Libra System [1] denoted as Libra. Finally, we include in our experiments a state-of-the-art cluster-based CF algorithm [4] denoted as DM. Our experiments are performed with the 100K MovieLens real data set, which consists of 100,000 ratings assigned by

[1]http://delab.csd.auth.gr/MoviExplain



Our Justified Recommendations

[Movie id]	[Movie title]	[The reason is]	[because you rated]
1526	Witness (1985)	Ford, Harrison (I)	21 movies with this feature
1273	Color of Night (1994)	Willis, Bruce	7 movies with this feature
1004	Geronimo: An American Legend (1993)	Hackman, Gene	7 movies with this feature
1442	Scarlet Letter, The (1995)	Oldman, Gary	7 movies with this feature
1044	Paper, The (1994)	Close, Glenn	7 movies with this feature
693	Casino (1995)	De Niro, Robert	6 movies with this feature
274	Sabrina (1995)	Pollack, Sydney	6 movies with this feature
1092	Dear God (1996)	Kinnear, Greg	5 movies with this feature

Figure 2: Explaining Recommendations

943 users on 1,682 movies. The range of ratings is between 1(bad)-5(excellent). The extraction of the content features has been done by joining with the contents of the internet movie database (imdb) and selecting 3 different classes of features: genres, actors, and directors. The join process yielded 23 different genres, 1,050 directors and 2,640 different actors and actresses. In the following experiments, the default size of the recommendation list, N, is set to 20, the neighborhood size k, is set to 10 (after tuning), and the size of the training set is set to 75%.

4.1 Evaluating Recommendations and Explanations

To measure the accuracy of recommendations, we use the well known measures of precision and recall. Precision and recall are defined as follows:

- *Precision* is the ratio of R_L to N.
- *Recall* is the ratio of R_L to R,

where N denotes the size of the recommendation list L, R_L denotes the number of relevant items that are included in L, and R denotes the total number of relevant items.

Precision and recall concern only the rating profile of a user u and measure the accuracy of L. However, precision and recall cannot distinguish between a relevant item from a more relevant item. To cope with this problem and to measure the quality of the justification, we introduce a user-oriented measure, called *explain coverage*.

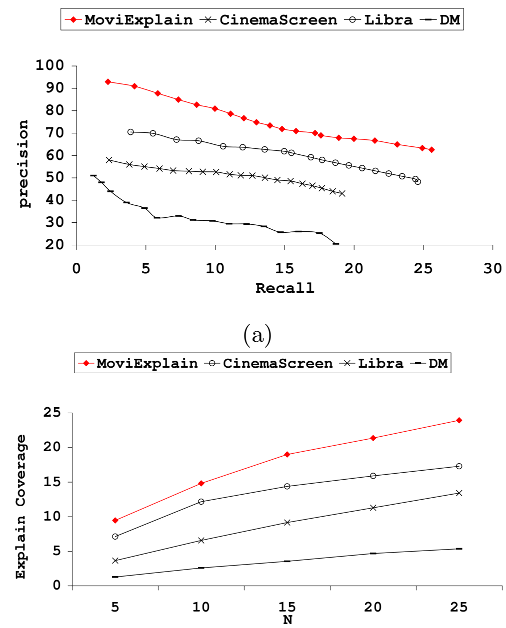
For a user u that receives a recommendation list L, the *explain coverage* for the justification list J is defined as follows:

$$Explain\ \ coverage(u, J) = \frac{\sum\limits_{\forall (f_i, c_{f_i}) \in J} \min\{c_{f_i}, P(u, f_i)\}}{\sum\limits_{\forall f_i \in F} P(u, f_i)}, \qquad (1)$$

where each pair (f_i, c_{f_i}) denotes that feature f_i has overall frequency c_{f_i} inside L and $P(u, f_i)$ is the frequency of f_i in the feature profile of u. *Explain coverage* takes values in the range [0, 1], whereas values closer to 1 correspond to better coverage.

4.2 Measuring Precision, Recall, and Explain Coverage

First, we compare the four algorithms by measuring precision vs. recall. Figure 3a plots the precision-recall diagram for the four algorithms (precision and recall are given as percentages). In particular, to obtain varying precision-recall values, we varied the number of the recommended movies (i.e., the parameter N). As expected, MoviExplain attains the best precision in all cases. The reason is two-fold, MoviExplain takes into account the duality between users and items by using biclustering, and, moveover, it detects partial matching of users' preferences.



(b)

Figure 3: Comparison between MoviExplain, CinemaScreen, Libra and DM in terms of (a) precision vs. recall and (b) *explain coverage* vs. N.

Next, we compare the four approaches in terms of *explain coverage* vs. the size N of the recommendation list. The results are presented in Figure 3b (*explain coverage* is given as percentage). MoviExplain outperforms the other methods in all cases. The reason is that MoviExplain uses groups of users, whereas the other methods are based solely on individual users.

4.3 User Study

We conducted a survey to measure user satisfaction against the three styles of explanations: "keyword" style (denoted

Expl. Styles	μ_r	σ_r	μ_d	σ_d	$Corr$	μ_p	σ_p
KSE	3.70	0.55	0.46	0.13	-0.10	1.86	1.02
ISE	3.97	0.63	0.73	0.14	0.13	2.26	1.20
KISE	3.30	0.56	**0.06**	0.13	**0.25**	**3.71**	1.08

Table 1: Results of the user survey.

as KSE), "influence" style (denoted as ISE), and our style of explanation (denoted as KISE), which combines the two aforementioned ones. We designed the user study with 42 pre- and post-graduate students of Aristotle University, who filled out an on-line survey, following a procedure that is similar to the one in Bilgic and Mooney work [1].

The survey was conducted in three steps (more details can be found in [1]): Firstly, we asked each target user to provide our system with ratings for at least five movies, so that a decent recommendation along with some meaningful explanations could be provided. Secondly, we asked them to rate separately, from 1 (dislike) to 5 (like), each recommended movie based on the three different styles of explanations (these ratings are denoted as Explanation ratings). This rating has been done after we had removed the titles of the recommended movies, because we did not want the target users to be influenced by them. Thirdly, the target users rated again each recommended movie (this rating is denoted as Actual rating), after they had seen the hidden information about it. If we accept that a good explanation lets the user accurately assess the quality of the movie, the explanation style that minimizes the difference between the ratings provided in the second and the third step is the best. Moreover, after we conducted the survey, we asked target users to rate separately each explanation style to explicitly express their actual preference among the three styles.

We assume that, (1) KISE will allow users to accurately estimate ratings better than KSE and ISE and (2) that KISE will be the users' favorite choice, because it is more informative and combines the other two explanation styles.

Our results are illustrated in Table 1. The second and third columns contain for each explanation style, the mean μ_r and standard deviation σ_r of the ratings provided by users in the second step of the survey (Explanation ratings). Regarding the third step of the survey, the mean value of the Actual ratings was 3.24, whereas the standard deviation of the Actual ratings was 0.45.

As earlier described, the best explanation is the one that allows users to best approximate the Actual rating. That is, the distribution of difference between Explanation ratings and Actual ratings should be centered around 0. We measured the mean μ_d and standard deviation σ_d of the differences between Explanation ratings and Actual ratings. These values, for each explanation style, are presented in the fourth and fifth columns of Table 1. KISE has the smallest μ_d value equal to 0.06. We run paired t-tests with the same null hypothesis $H_0(\mu_d = 0)$ for all three styles. We found that for KISE $H_0(\mu_d = 0)$ is accepted at the 0.01 significance level. In contrast, for KSE and ISE we reject $H_0(\mu_d = 0)$ at the same significance level. This verifies our first (1) assumption.

We also calculated Pearson Correlation (denoted as $Corr$) between Actual and Explanation ratings, to show that the Actual and Explanation ratings follow similar patterns. The results are presented in the sixth column of Table 1. KISE has positive correlation with Actual rating, equal to 0.25. This also supports our first (1) assumption, because it shows that the Actual and KISE ratings are positively correlated.

Finally, the last two columns of Table 1 present the mean μ_p and standard deviation σ_p of ratings provided by the users to explicitly express their preference for each explanation style. KISE attained a μ_p value equal to 3.71 (in 1 to 5 scale), which is the largest among all styles. We run paired t-test, and found out that the difference of KISE from KSE and ISE is statistically significant at the 0.01 level. This supports our second (2) assumption.

5. CONCLUSIONS

The need of providing justifiable recommendations has recently attracted significant attention, especially in e-commerce sites (Amazon, e-Bay etc.). In this paper, we proposed MovieExplain, a movie recommender system that goes far beyond just recommending movies. It attains both accurate and justifiable recommendations, giving the ability to a user, to check the reasoning behind a recommendation. In the future, we intent to use in MoviExplain also natural language processing to provide more robust explanations.

6. REFERENCES

[1] Bilgic, M. and Mooney, R.J. Explaining Recommendations: Satisfaction vs. Promotion. In *Proceedings of the Recommender Systems Workshop (IUI Conference)*, 2005.

[2] Billsus, D. and Pazzani, M. A personal news agent that talks, learns and explains. In *Proceedings of the Autonomous Agents Conference*, pages 268-275, 1999.

[3] Herlocker, J. and Konstan, J. and Riedl, J. Explaining collaborative filtering recommendations. In *Proccedings of the Computer Supported Cooperative Work Conference*, pages 241-250, 2000.

[4] Jin, R. and Si, L. and Zhai, C. A study of mixture models for collaborative filtering. *Information Retrieval*, vol. 9, issue 3, pages 357-382, 2006.

[5] Melville, P. and Mooney, R. J. and Nagarajan, R. Content-Boosted Collaborative Filtering for Improved Recommendations. In *Proceedings of the AAAI Conference*, pages 187-192, 2002.

[6] Mooney, R. and Roy, L. Content-based book recommending using learning for text categorization. In *Proceedings of the ACM DL Conference*, pages 195-204, 2000.

[7] Salter, J. and Antonopoulos, N. CinemaScreen Recommender Agent: Combining Collaborative and Content-Based Filtering. *Intelligent Systems Magazine*, vol. 21, issue 1, pages 35-41, 2006.