Comparing Feature-based and Clique-based User Models for Movie Selection

Joshua Alspector and Aleksander Kolcz
Department of Electrical and Computer Engineering
University of Colorado at Colorado Springs
1420 Austin Bluffs Pkwy.
Colorado Springs, CO 80918, USA
E-mail: {josh, ark}@eas.uccs.edu

Nachimuthu Karunanithi
1F-319B, Bellcore,
445 South Street
Morristown, NJ 07960, USA
E-mail: karun@bellcore.com

ABSTRACT
The huge amount of information available in the currently evolving world wide information infrastructure at any one time can easily overwhelm end-users. One way to address the information explosion is to use an "information filtering agent" which can select information according to the interest and/or need of an end-user. However, at present few such information filtering agents exist. In this study, we evaluate the use of feature-based approaches to user modeling with the purpose of creating a filtering agent for the video-on-demand application. We evaluate several feature and clique-based models for 10 voluntary subjects who provided ratings for the movies. Our preliminary results suggest that feature-based selection can be a useful tool to recommend movies according to the taste of the user and can be as effective as a movie rating expert. We compare our feature-based approach with a clique-based approach, which has advantages where information from other users is available.

KEYWORDS: user modeling, information filtering, collaborative filtering, feature extraction, neural networks, linear models, regression trees, bagging, CART.

INTRODUCTION
In recent years, computer-network-based information services have gained wide acceptance as evidenced by the explosion of information utilities and services on the World-Wide Web. The huge amount of information available can easily overwhelm end-users, a situation that is likely to worsen in future multimedia information networks unless the end-user has the ability to filter information based on its relevance.

An important part of an information filter is a user model to predict a target user's preferences. In this study we concentrate on one particular type of the user modeling problem involving movie selection. In the future video-on-demand services may well be able to supply customers with any movie desired. Considering the very large number of potential choices, the problem of selecting a movie conforming with the user's taste will certainly be of importance, and indeed the problem has recently received considerable attention both in the commercial [2-5] and academic world [1, 13, 14, 19].

The problem of designing a movie rating system tailored to the preferences of a particular user is approached from two angles. In the first model, we use a collaborative filtering approach [12], where a "clique" of users whose taste is similar to that of the target user is found. Subsequent predictions of the target user's rating of a particular movie are made by aggregating the ratings obtained from the clique members for that movie.

In the alternative approach, a set of movie features is used to create a trainable network model to provide rating predictions. Several types of network architectures (including linear and nonlinear models) are investigated for that purpose.

In this study, we evaluate and compare the use of clique and feature-based movie recommendation systems for video-on-demand service. The study is based on a feature database of 7389 movies. We evaluate our approaches on the data collected from 242 voluntary subjects who provided ratings for the movies. Out of this user group 10 subjects were selected as the target users on which the effectiveness of our user models was tested. The obtained results indicate very good performance levels for the clique models and suggest that the feature-based selection can be a useful tool (outperforming a movie rating expert) to recommend movies according to the taste of the user.

AN EXPERIMENT ON MOVIE SELECTION

Proposed Approach to Video Selection
A schematic representation of the proposed filtering system is shown in Figure 1. The system incorporates aspects of both information filtering [17] and information retrieval [9]. Namely, a user may be interested in watching a certain type of a movie at a particular time. At the same time the system might provide the user with periodic suggestions regarding new movie releases as they become available. Moreover,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Digital Libraries 98 Pittsburgh PA USA
Copyright ACM 1998 0-89791-965--3/98/ 6...$5.00

META-1037
IPR2025-01248

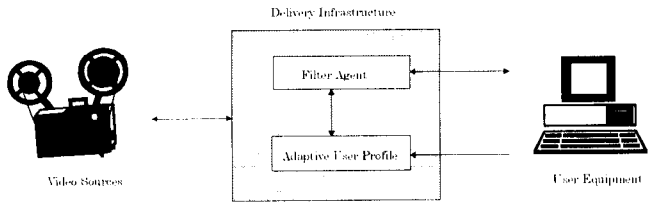

Figure 1: An abstract view of movie selector.

the movie data sources will probably be organized in a structured way to facilitate retrieval, but the information about particular movies may be available from various distributed sources.

Our model assumes that both the filtering agent and the user profile may be set, controlled, and maintained either by the delivery infrastructure, on behalf of individual users, or by the individual end-user. The feedback can be "passive", "active", or a combination of both. In passive feedback, the user need not participate; rather the actions of the user will be monitored and stored. For example, the user may stop the movie in the middle, or try to fast-forward in the middle, or see the entire movie and try to re-dial for another run. In active feedback, the user will be requested to respond to one or more relevant questions. This, for example, may include whether the user liked the movie or not, what is the rating for the movie, etc. In our initial implementation we incorporate an active feedback mechanism.

Within this framework, several alternative filtering approaches can be envisioned for the movie recommendation application. In this study, we evaluate two fundamentally different approaches: a *clique-based* approach (The *clique-based* approach is a variant of the approach suggested by Will Hill [13] of Bellcore. It has similarities to the "stereotype" of Rich [20], the "communities" of Orwant [18], and "social filtering" as used by Maes [16].) and a *feature-based* approach [14]. In the *clique-based* approach, movies are recommended using the ratings of a set of users who might have similar taste (according to some suitable metric). In the *feature-based* approach, first a model is built using a set of important features of the movies that a user has seen and rated, and then that model is used to predict the ratings for movies that the user wants to see. Here, the recommendation is based on the movie content (e.g., represented by its features). Although both methods require active user participation (to provide the movie ratings), it should be noted that we are comparing two distinct and extreme data-driven approaches and would probably recommend combining these with a profile-based approach for a more flexible system [8].

A Clique-Based Approach

The clique-based approach is based on the hypothesis that the average rating of a *clique* of users is the best indicator of an individual's future rating. The approach of using other people opinions to determine the quality/relevance of an information item (or a product) is generally known under the name of collaborative filtering [12, 15]. This approach is currently gaining in popularity due to the widespread use of the internet, where it is relatively easy to poll the opinion of many users in a relatively short time.

A set of users form a clique if their movie ratings are closely related. Each user for whom we wish to predict ratings has a unique clique composed of other users whose ratings are similar. The members of the clique who have rated a movie that the target user has not seen predict the rating of the target user for that movie. As a similarity measure, we use the Pearson correlation coefficient (which has been applied in the context of user modeling by Allen [6, 7]), C, which is a normalized dot product of the vectors of the ratings of two users. For any two vectors **a** and **b**, the correlation similarity measure is defined as

$$C = \mathbf{a} \cdot \mathbf{b} / \left(\sqrt{\mathbf{a} \cdot \mathbf{a}} \cdot \sqrt{\mathbf{b} \cdot \mathbf{b}} \right) \qquad (1)$$

The values of C range from -1.0 to +1.0 and, generally, any positive correlation indicates a similarity of taste, but in order to maximize the rating-prediction performance, it is desired to select a group of users whose positive correlation with the target user is greatest.

A member of the clique for whom we want to predict a future rating is considered a *target user*. In order to identify a clique for a target user, we define two parameters: C_{min}, the *correlation threshold* and S_{min}, the *size threshold*. The parameter C_{min} defines the minimum correlation required for a user to become a member of the clique of a target user. Thus, the parameter C_{min} imposes a minimum limit on similarity of ratings required for a user to be a member of the clique of a target user. The second parameter, S_{min}, defines a lower limit on the number of movies that a user must have seen (in common with the target user) and rated in order to be a member of the clique. The parameter S_{min} is used to restrict users who have not seen the same movies that the target user has seen from entering the clique. Thus, a clique is defined for a target user by specifying suitable values for C_{min} and S_{min}.

In our current implementation, we use a constant value of 10 for S_{min} and a positive variable value for C_{min} such that the number of users in a clique is held constant at 40. Note that if S_{min} is too small, that rater is unlikely to help in predicting a movie that a target user has not seen. Note also that if C_{min} is too low, there will be raters whose correlation with the target user is not strong, and so prediction is also compromised.

Thus, the movie recommendation algorithm for the clique-based approach involves the following steps.

1. Initialize parameters C_{min} and S_{min}.
2. Identify a clique for the target user.
3. Estimate ratings for the movie that the target user wants to see by considering the ratings by the members of the clique who have already rated that movie.

We considered two approaches of aggregating the ratings of clique members (i.e., step 3). One straightforward implementation is to use a simple arithmetic mean of the ratings of the members of the clique, i.e.,

$$r(m) = \frac{\sum_{i=1}^{N} c_i(m)}{N} \tag{2}$$

where $r(m)$ represents the clique rating of a particular movie, m, N is the size of the clique, and $c_i(m)$ is the rating of movie m given by the ith clique member. Thus, the average rating of the clique becomes the predicted rating for the target user. A more sophisticated strategy assigns the rating of each clique member a weight corresponding to the average correlation between ratings of the clique member and the target user. The output of a correlation-ranking predictor is given by

$$r(m) = \sum_{i=1}^{N} w_i \cdot c_i(m) \tag{3}$$

where $w_i(m)$ is the normalized correlation (weight) between the ith clique member and the target user.

A Feature-Based Approach

The feature-based approach rests on the notion that the features of the movies can be useful in recommending movies. This conforms with the standpoint of content-based information filtering, where it is assumed that the degree of relevance (to a particular user) of a piece of information can be determined by its content.

Movies (and other video sources) represent much more complex material than text documents and the problem of extracting their relevant features is certainly challenging. However, important indirect data (i.e., outside the actual movie video) characterizing any movie are readily available. Some of the features that can be used to recommend movies include MPAA ratings (e.g., PG), expert critic ratings, movie category (e.g., drama), name of the director, leading actors/actresses, and awards received.

The algorithm for the feature-based approach is as follows:

1. Extract relevant features from the movies that the user has rated.

2. Build a model for the user by associating selected features (as inputs) and the ratings (as output).

3. Estimate ratings for the movie that the target user wants to see by considering its features as new input to the model.

To implement this algorithm, one must select a proper set of features and a correct model to predict ratings. We already suggested several useful features for rating movies. Some of them are readily available once the movie has been released (e.g., the director), while others (e.g., MPAA or Category) rely on external assignment and can themselves exhibit a significant variation. The problem of selecting the most representative set of movie features certainly remains open. In this study, we use only a set of seven features explained below (in cases where a short version of the feature name is used later, the short version is provided in parentheses):

- *Category*: (comedy, drama, etc.). The total of 25 categories was considered with 1-of-N unary representation.

- *MPAA rating* (MPAA): official rating (with 6 possible values) given in the USA to movies approved for distribution; 1-of-N unary representation was used.

- *Maltin rating* (Maltin): the ratings (on the scale from 0 to 4 normalized to the [0,1] range) of professional reviewer Leonard Maltin.

- *Academy Award* (AA): (won =1), (nominated =0.5) or (not considered =0).

- *Length*: For each movie its length (in minutes) was normalized with respect to the overall average movie length in the database.

- *Origin*: made in the USA (Origin=1), made in the USA with foreign collaboration (Origin=0.5) and foreign made (Origin=0).

- *Director*: This feature was pre-processed by obtaining an average rating by that user for movies by each particular director. In the case where the user has seen no movies by a particular director, the value of this feature was set to the overall average rating of movies seen by that user.

Although in this study we concern ourselves with features based on numerical and textual information associated with movies, one could also attempt to extract explicit features based on the video content. Certainly, feature sets alternative to the one used here could be investigated. Each of these features was normalized to take an integer value in $\{0,1\}$ or a real value in $[0,1]$.

Admittedly, the set of features chosen may not be sufficient to describe the attractiveness of a movie from the point of view of a target user. The rationale for selecting a small set of features is that we wanted to evaluate whether we can build a reasonable model with as few features as possible. As far as the problem of selecting an appropriate model incorporating these features is concerned (i.e., step 2 in the strategy outlined), we do not make any specific recommendation as to what modeling approach might be superior. The choice for modeling, for example, may include a linear or nonlinear regression model, an expert system, a neural network, or other approaches. In this study, both a linear and nonlinear approach are illustrated.

A Linear Model. One of the most natural assumptions to make is that of a linear influence of each of the features

involved on the overall rating. Thus, after including a constant bias component a model of this type takes the form of:

$$r(m) = \sum_{i=1}^{N} w_i \cdot x_i(m) + b \qquad (4)$$

where $x_i(m)$ denotes the value of the ith feature for movie m, w_i is the weight associated with that feature, and b represents the bias. Creation of such a model is essentially equivalent to a multiple linear regression on the set of feature variables and the its solution can be obtained using least-squares techniques.

A Linear Model with Feature Grouping. Our experience with the linear model led to an observation that the MPAA and Category features represent very sparse (i.e., 1-of-N and few-of-N) encoding, which may not be best suited for solving the linear regression problem, as the very sparse encoding significantly adds to the dimensionality of the vectors and matrices involved. In fact, during simulations with the linear model the linear system proved to be ill-conditioned in most cases. As a result two pre-processing networks were implemented, one for each of the features mentioned. As the MPAA categories are represented strictly by 1-of-N encoding (i.e., for any movie only one MPAA category is assigned), a simple lookup-table model was created to obtain the average rating for movies in a given MPAA category (for each user). As far as the Category feature is concerned, movies can be assigned a number of different categories (e.g., action as well as adventure), and a lookup-table technique would require too many entries to account for all the possible category combinations. Therefore a separate linear network was created to provide the Category-based ratings. Thus processed, the outputs of the MPAA and Category features were subsequently fed, together with the remainder of the features, into a top-level linear network. The overall network architecture is depicted in Figure 2.

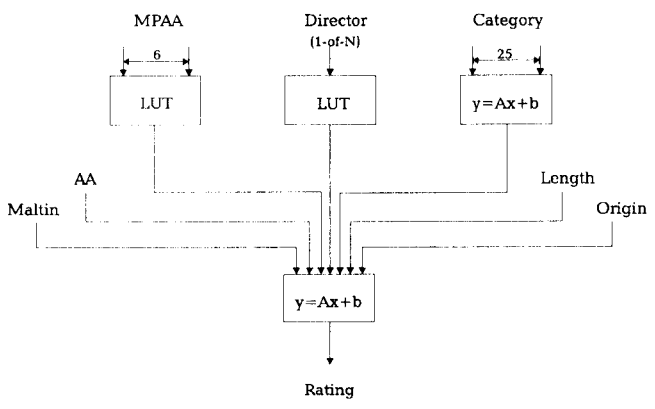

Figure 2: The linear network with MPAA and Category feature groupings. The (*y=Ax+b*) blocks correspond to linear-network modules, whereas the (LUT) blocks represent lookup tables. Wherever relevant, the original size/encoding of the features is indicated.

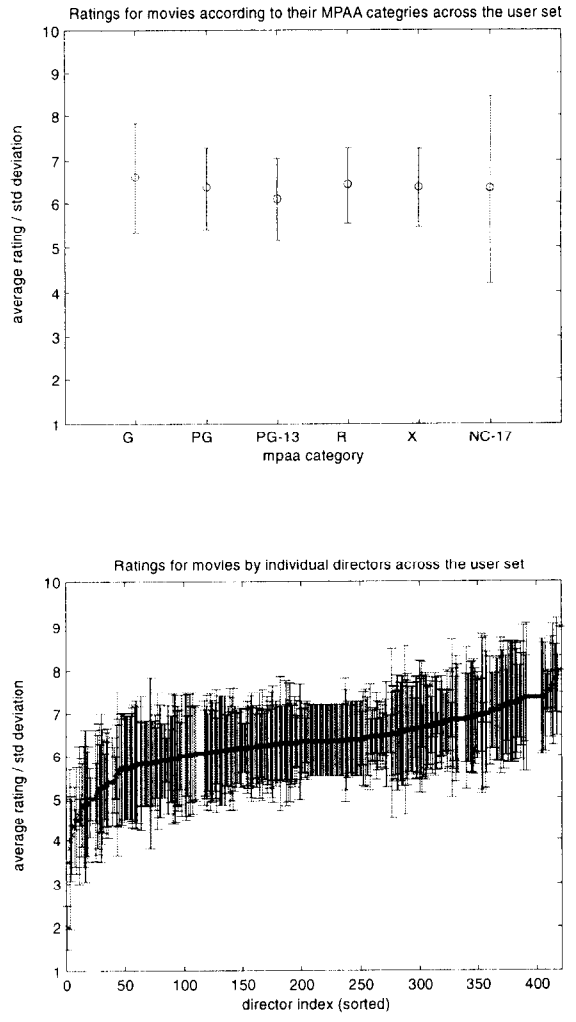

Figure 3: The average movie ratings (and their standard deviations) according to the MPAA categories (top) and movie directors (bottom). The bottom plot is ordered according to the average rating associated with movies by individual directors. The differentiation within the MPAA feature set appears to be much smaller than that in the directors set.

A Multiresolution Approach: A Priori Assignment of Feature Importance. It appears that some members of the feature set represent more average properties of the data than others. In particular, for any given set of user data, most of the MPAA categories will have many movies sharing that category, which makes differentiation of user ratings based on this feature quite broad. On the other hand, for each value of the Director feature, the number of movies sharing it will be quite small (usually one or two for the data sets considered here), which makes modeling based on this feature very data sensitive (and hence generalization is difficult). Figure 3 shows the variability of movie ratings (averaged across the user set) obtained for the MPAA (top of Figure 3) and Director (bottom of Figure3) features. In both cases the ratings were obtained via a lookup-table approach. It can be seen that for modeling with MPAA features only, there is

little difference for average ratings across the different categories. On the other hand, the set of directors can be clearly divided into those producing better and worse movies. Most of the directors are judged as average, but those responsible for very good and quite bad (especially those) movies are also identified. Taking these observations into account, it appears justifiable to attempt to divide the feature set into several layers (or groups), where each layer would represent a certain level of detail as far as rating prediction is concerned. The following ordering has been used (directed from low-detail to high-detail features):

[MPAA(low detail)]→[Category] →[Length, Origin, Maltin, AA] →[Director(high detail)]

Thus the network consists of four layers and its parameters were estimated through a multistage learning process where, at each stage, the parameters of one network layer were found.

A CART Network. Despite the nonlinear modeling elements (i.e., lookup tables) used to process a few of the features in the models described above, these architectures are predominantly linear. In order to enlarge the model space a purely nonlinear method was also considered. The CART (Classification and Regression Trees) [11] network was chosen in the hope that it would take advantage of the potential nonlinear dependencies between the individual feature variables. An important rationale for choosing this particular network was the fact that CART trees are also very useful as far as interpretation of network operation is concerned. An example of a CART tree grown on the user movie data is shown in Figure 4.

A CART Network with Bagging. Nonlinear models, such as CART (and neural networks), tend to be sensitive to the training data, which leads to a large variance component of the generalization error they produce. It has been recently noticed [10] that this effect can be largely overcome by aggregating a number of such networks, each created with a different variant of the training set. The different variants are generated by bootstrap sampling (i.e., sampling with repetition) from the original training set, and the term bagging (bootstrap aggregation) has been coined by Breiman [10] to describe the overall process. Bagging was applied to the CART network in order to reduce variance component of the generalization error.

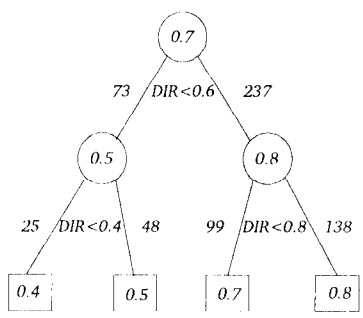

Figure 4: Example of a tree created for user U39; the DIR label corresponds to the Director variable (representing the average rating given to movies by a particular director by the target user); a number within a circle, or a rectangle, corresponds to the average rating of movies falling into the node; the numbers beside the connecting lines represent the numbers of training points following a particular split.

RESULTS

Data Collection

The feature database for the experiment was initially populated from the Microsoft Cinemania™ CD-ROM for 1548 movies in Will Hill's survey [13], and then expanded by data obtained from the Internet Movie Database to the current number of 7389 elements. The features collected include the set of seven described before, as well as leading actor/actress and a short review for each movie. As pointed out earlier, only the first seven features are used in this study. The test-case users are 242 internet subscribers who volunteered to rate the movies that they had seen. Each movie was rated on a scale of 1 to 10, with 1 being the worst, and 10 the best. These ratings were subsequently normalized to the [0,1] range. The number of movies rated by an individual user varied from 0 to a maximum of 460, with the average being 177. For the purpose of evaluation, we selected 10 users who have rated approximately 350 or more movies as our *target users*. They are labeled U# (U3, U21, U39, U41, U46, U77, U111, U124, U145, and U178).

User ID	# rated	# complete
U21	407	278
U77	356	250
U3	372	274
U41	360	253
U46	460	325
U111	382	263
U145	374	262
U124	427	292
U39	440	310
U178	410	291

Table 1: The numbers of movies rated (second column) by individual target users shown together with the respective numbers of movies for which all features were actually available (third column).

Unfortunately, not all of the movies rated by the target users had all their features present in the currently available database. Since we did not concern ourselves with the treatment of missing values in our feature-based models, only a reduced set data could be used in the experiments relying on the use of movie features, although the clique-based methods could use the whole data set of user ratings (since no additional movie features were required). Table 1 shows the numbers of movies rated by users from the target groups (middle column), as well as the numbers of movies rated for which all features were available (bottom column). The latter set will be referred to as the *reduced* data set.

In order to understand whether there is any obvious relationship among the target users and others in the sample, a preliminary analysis was performed on the ratings of each target user against all others in the sample. Our analysis shows that there is no strong correlation between a particular target user and the rest of the sample. However, a selected clique of users, as previously described, should have strong correlation as we will see. We also noticed that if a user sees as many of the movies as the target user, there is often a strong correlation between them. Note furthermore that the sample of movie viewers that we have chosen to analyze may not be representative of the broad population since they were willing to rate movies over the internet and have seen many more movies than the average viewer.

Experimental Setup

The experiment for this study was conducted as follows: For each target user, we split the data set into a *training set* and a *test set*. The training set was used to build the model while the test set was used to validate the model. (In the clique-based approach, the training set was used to identify the clique for the target user.) The training set had 90% of the ratings and the test set had the remaining 10% of the data. In order to make a fair comparison, we used the standard statistical technique of cross-validation by splitting the data into 10 different mutually exclusive training and test sets. Thus, for each target user there were 10 experiments whose results were then aggregated to produce the overall performance measure (in the form of the Pearson correlation coefficient) for each of the models considered.

The Clique-Based Approach

The clique method was applied both to the whole and to the reduced data set available, with the latter considered primarily for better comparison with the feature-based approach. In either case, once the clique was identified, its ratings were obtained both by simple averaging (Eq. 2) and by weighted averaging (i.e., correlation ranking) (Eq. 3). Table 2 summarizes the results obtained using the clique method. The bottom-most row corresponds to the average performance across the user set. It can be seen that there is very little difference between the performance of equal-ranking (i.e., simple averaging) and correlation ranking methods. In fact, for the reduced-set experiments, the correlation ranking was marginally worse, whereas for the whole-data experiments it was marginally better. The equivalence of the ranking methods considered was due to the fact that there was little differentiation within the set of correlations between each target user and the members of its clique. With different user data the distinction between these two ranking types could probably be more visible.

On the whole, the clique method performed very well, although it is interesting to see that the performance obtained on the reduced data was slightly better than the performance given by the cliques built using the complete data. This may indicate a case of overfitting (common to many data-driven models), where some marginally better correlation performance of certain users, on smaller sets of movies in common with a target is more likely to be picked up in a larger data set. Such behavior could probably be avoided if a more sophisticated method of clique design was chosen (recall that in the method considered here users who rated as few as 10 movies in common with the target could become members of a clique). For comparison with the feature-based approach, the reduced-data clique predictor with correlation ranking was used.

The Feature-Based Approach

Analysis of the CART Networks. A typical single CART tree grown on user data (user 39 in this case) is shown in Figure

User ID	Clique	Clique w. ranking	Clique reduced	Clique (red) w. ranking
U21	0.37	0.38	0.68	0.67
U77	0.25	0.26	0.36	0.34
U3	0.48	0.48	0.40	0.39
U41	0.38	0.38	0.59	0.59
U46	0.57	0.57	0.64	0.65
U111	0.70	0.71	0.84	0.84
U145	0.47	0.48	0.30	0.31
U124	0.77	0.77	0.46	0.46
U39	0.66	0.67	0.76	0.75
U178	0.80	0.80	0.83	0.83
average:	0.55	0.55	0.58	0.58

***Table 2: Performance of the clique based models; the* bottom row corresponds to the average performance across the user set. Correlations can vary between -1 and 1, where the correlation of 0 implies no systematic relationship between *the ratings provided by the user* and its model, while the correlation of 1 represents the desired target performance.**

4. It is seen that all of the splits occur on the Director variable, which indicates high relevance of this variable as far as the explanation of the variation in user rating is concerned. However, when applied to a test set, such trees tend to perform poorly, as the value of the Director feature for points in the test set has to be set on the basis of the training set, and relatively few movies in the user data share directors. Consequently, most values of this features for points in the test set are equal to the average movie rating of the particular user, so there is little differentiation for the ratings generated for the test set (as most of them fall into the same terminal node of the tree). This behavior suggested that the method of setting the Director feature should be modified in order to facilitate model design. It was chosen that the value of this feature should represent an average rating of movies directed by a given director if such number is greater than one. Otherwise the value of the feature was assigned the average rating of movies seen by the user. This improved the generalization performance of CART and all other methods considered. The tree grown for user 39 on the modified data is shown in Figure 5. It is seen that a much smaller tree results, although the splitting is still done on variable Director. Although splits on other variables also occurred in some cases, the tree models used predominantly the Director feature, which suggests that the remaining variables provide only a weak explanation of the variability of user ratings (i.e., more informative feature sets should be sought after).

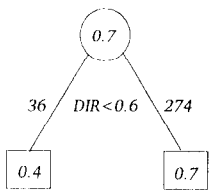

Figure 5: Example of a tree created for user U39 with the modified encoding of the Director feature (compare with Figure 4). Since the variability of this feature is significantly decreased a much smaller tree results.

Performance Comparison. Table 3 shows the combined results of our study. The results are presented in terms of the correlation coefficient (Eq. 1) between the actual ratings by the target users and the ratings by different movie recommendation approaches, where a higher correlation implies that the predicted ratings are close to the actual ratings. Table 3 summarizes the results of the feature based methods and compares them with the clique-based approach (i.e., with the correlation-ranking variant of the clique model). The Maltin feature (second column in Table 3) is also added to assess to what extent user-dependent modeling improves over an independent expert. The bottom row of the table shows the correlation performance averaged across the user set.

It can be seen that the clique-based method led to significantly better results (with the exception of users U77 and U145) than the ones obtained with the feature-based modeling, although most of the feature-based networks provided an improvement over the Maltin rating (a movie rating expert). These results should not be over-generalized however, since other data sets and, indeed, different feature-based modeling approaches could reverse the performance outcome. Of the feature-based networks considered, the linear-type networks seem more appropriate for this problem than the CART-based networks. All feature-based networks except the CART-based networks rated better than Maltin's ratings. The fact that the globally linear network led to the poorest results (within the "linear" group) can probably be attributed to the numerical instability (i.e., ill-conditioning of linear systems) encountered in computations for this case. By introducing sub-networks for grouping the MPAA and Category features (column 4), and by introducing a hierarchical structure to the network (column 5) the instability of the linear systems can be avoided and the performance levels are improved. It can be seen that bagging leads to a definite improvement over a single CART network. The generally poorer performance of this network type is due to overemphasizing the Director feature during tree creation (described before), as this feature carries disproportionally less information in a test set than in a training set. Without modifying the encoding of this feature the performance would be worse (also for the linear

User ID	Maltin rating	Lin.	Lin./ grp	Multi resol.	CART	CART/ ba.	Cliq./ rank
U21	0.07	0.35	0.36	0.34	0.26	0.31	0.67
U77	0.15	0.32	0.37	0.29	0.25	0.32	0.34
U3	0.27	0.23	0.21	0.22	0.19	0.21	0.39
U41	0.32	0.35	0.39	0.40	0.20	0.26	0.59
U46	0.33	0.47	0.46	0.44	0.29	0.36	0.65
U111	0.54	0.49	0.53	0.50	0.25	0.38	0.84
U145	0.31	0.14	0.31	0.34	0.10	0.20	0.31
U124	0.38	0.35	0.31	0.31	0.14	0.23	0.46
U39	0.36	0.50	0.54	0.50	0.36	0.44	0.75
U178	0.36	0.33	0.32	0.30	0.15	0.29	0.83
avg:	0.31	0.35	0.38	0.36	0.22	0.30	0.58

Table 3: Correlations of predicted vs. actual ratings by different methods; the bottom row corresponds to the average performance across the user set. Correlations can vary between -1 and 1, where the correlation of 0 implies no systematic relationship between the ratings provided by the user and its model, while the correlation of 1 represents the desired target performance.

methods). However, these results suggest that additions should be made to make to the selected set of features (e.g., the leading actor/actress features).

CONCLUSIONS

We developed several feature-based movie rating systems and compared them against an expert and two variants of the clique-based approach. The model set considered extends (and agrees with) our previous results involving a neural network model [14]. Our preliminary results, based only on a few important features, suggest that the feature-based approach can be used for information products where there are no other raters. Thus, the feature-based approach is useful for new products or where privacy for market research is an issue. The clique-based approach, on the other hand, is advantageous if enough information about other users is available.

REFERENCES

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

[2] http://rw.cinemax.com/critic

[3] http://vguide.sepia.com

[4] http://www.moviecritic.com

[5] http://www.moviefinder.com

[6] R. B. Allen, "Mental Models and User Models," in *Handbook of Human-Computer Interaction*, M. Helander, T. K. Landauer, and P. Prabhu, Eds., 2 ed: Elsevier Science, 1997, pp. 49-63.

[7] R. B. Allen, "User models: theory, method, and practice," *International Journal on Man-Machine Studies*, vol. 32, pp. 511-543, 1990.

[8] M. Balabanovic and Y. Shoham, "Fab: content-based collaborative recommendation," *Communications of the ACM*, vol. 40, pp. 66-72, 1997.

[9] N. J. Belkin and W. B. Croft, "Information Filtering and Information Retrieval: Two Sides of the Same Coin?," *Communications of the ACM*, vol. 35, pp. 29-38, 1992.

[10] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123-140, 1996.

[11] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*: Wadsworth, 1984.

[12] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using Collaborative Filtering to Weave the Information Tapestry," *Communications of the ACM*, vol. 35, pp. 61-70, 1992.

[13] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and evaluating choices in virtual community of use,," Proceedings of the Conference on Human Factors in Computing Systems (CHI'95), 1995.

[14] N. Karunanithi and J. Alspector, "Feature-Based and Clique-Based User Models for Movie Selection," Proceedings of the Fifth International Conference on User Modeling, Kailua-Kona, HI, 1996.

[15] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "GroupLens: applying collaborative filtering to Usenet news," *Communications of the ACM*, vol. 40, pp. 77-87, 1997.

[16] P. Maes, "Agents that Reduce Work and Information Overload," *Communications of the ACM*, vol. 37, pp. 31-40, 1994.

[17] D. W. Oard, "The State of the Art in Text Filtering," *User Modeling and User-Adapted Interaxrion*, vol. 7, pp. 141-178, 1997.

[18] J. Orwant, "Heterogeneous Learning in the Doppelganger User Modeling System," *User Modeling and User-Adapted Interaction*, vol. 4, pp. 107-130, 1995.

[19] B. Raskutti, A. Beitz, and B. Ward, "A Feature-based Approach to Recommending Selections based on Past Preferences," *User Modeling and User-Adapted Interaxrion*, vol. 7, pp. 179-218, 1997.

[20] E. Rich, "Users are Individuals: Individualizing User Models," *Int'l Journal of Man-Machine Studies*, vol. 18, pp. 199-214, 1983.

View publication stats