Feature-based and Clique-based User Models for Movie Selection: A Comparative Study

Joshua Alspector[*], Aleksander Kołcz[†] and N. Karunanithi[‡]

January 13, 1998

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 information filtering agents exist for the evolving world wide multimedia information infrastructure. 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.

1 Introduction

In recent years, computer-network-based information services have gained wide acceptance both within commercial and non-commercial sectors. This is evidenced by the explosion of information utilities and services on the World-Wide-Web portion of the internet. The information content in such services is mostly textual. However, the currently evolving internet is expected to support not only a variety of text-based information services but also various multimedia (hypertext, audio and video-based) information services. Some of the potential application domains in which the information infrastructure is likely to have

[*] ECE Dept., University of Colorado, Colorado Springs, CO 80918. email:josh@eas.uccs.edu

[†] ECE Dept., University of Colorado, Colorado Springs, CO 80918. email: ark@eas.uccs.edu

[‡] 1F-319B, Bellcore, 445 South Street, Morristown, NJ 07960. email: karun@bellcore.com

impact are: banking at home, access to electronic libraries, distance learning and laboratories, delivery of news and entertainment on demand, electronic shopping malls, law enforcement and security alertness, legal services, national health care and weather services, and telecommuting. Thus, the internet has the potential to change the way we work, communicate, travel, and generally access information.

The huge amount of information available in the information infrastructure at any one time can easily overwhelm end-users. Even within existing computer-network-based information services, providing information that is of interest to a particular end-user is not an easy task. For example, filtering relevant e-mail information in the internet is not easy because a single message may be sent over a set of mailers (e.g., filtering out messages that had passed through a particular server may stop many desired messages and still allow "e-mail spam" to arrive via a different route), a message may consists of a "thread" (a sequence of "replies" to the original mail), or the header may not reflect the actual content. This situation is likely to worsen in future multimedia information networks unless the end-user has the ability to filter information based on what is relevant to him/her.

Several useful text-based tools exist for navigational purposes (Obraczka, Danzig, & Li 1993) on the internet. An example of a common-to-all user interface for the internet is Mosaic, Netscape(TM) or Microsoft Internet Explorer(TM). These are hypertext-based easy-to-use interfaces built on top of various internet navigational and browsing tools such as Gopher, WAIS and World Wide Web, and incorporating search/retrieve services such as Archie and FTP. Even with such a common interface, these navigational aids require network support, and active participation of the end-user. Moreover, despite rapid progress, at present few equivalent filtering systems exist for the evolving multimedia information infrastructure.

An important part of an information filter is a user model to predict what a target user would like to filter. In the most straightforward approach, the users may be required to state their preferences in a more or less structured way (e.g., by way of creating a personalized profile). However, such an approach has obvious limitations and what is really desired is a system capable of modeling a particular user's preferences (or taste) on the basis of the actual choices and decisions made by the user during the course of his/her interaction with the information provider. In this study we concentrate on one particular type of the user modeling problem involving movie selection. This is a relevant problem considering that the data on the majority of movies made so far is being made available over the internet (e.g., the Internet Movie Database[1]), and 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. Many other applications of recommendation systems (e.g., music selection) have been reported in the literature (Maes 1994).

[1] http://www.imdb.com

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 (Goldberg et al. 1992), 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.

Not considered, but relevant to any useful user model is a third component based on a profile of the user’s interests, where some of the user’s preferences would be specified in a structured manner. An example is the “.newsrc” file for accessing USENET newsgroups, or the profile used to extract information of interest from the internet (to appear as a screen saver on the user’s computer screen) used by the Pointcast service. Such a profile is easily modified by the user and can be quite effective for filtering certain types of information (e.g., daily performance of the stock market). A useful information filter will likely combine all these approaches, and successful attempts to create such hybrid systems have been reported (Balabanović & Shoham 1997). What we envision is, in effect, “an adaptive user profile”, as will become clear.

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 these 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.

2 Information Filtering vs. Information Retrieval

Before we discuss information filtering approaches, we must make a distinction between information filtering and information retrieval methods. At an abstract level information retrieval and information filtering are two sides of the same coin because both are concerned with getting information to people who want it. Both information retrieval and information filtering applications are designed to deal with semistructured and unstructured data (e.g., text documents). Most textual information falls under the category of unstructured data and often their meaning is difficult to represent in a typical database (the syntax and the semantics of the fields of a data item are well defined). For example, in a semistructured *e-mail* message only header fields need to conform to certain standards. Additionally, information filtering systems need to deal not only with unstructured textual data but also with other types of data such as images, video

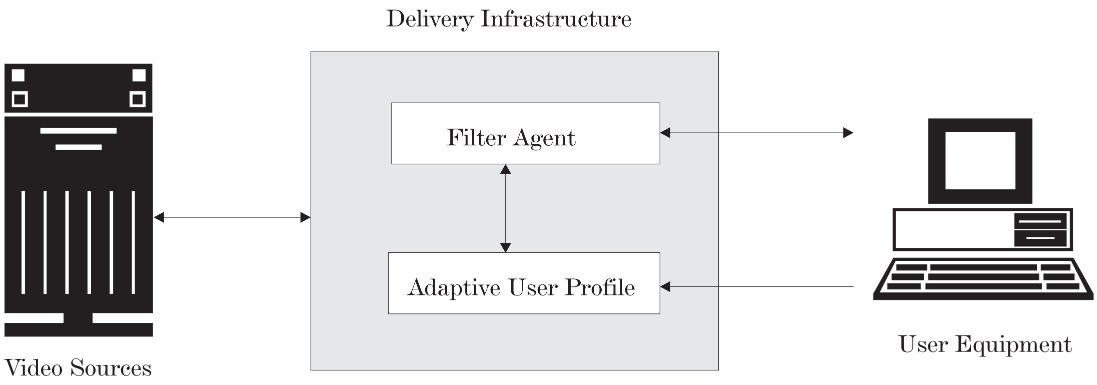


Figure 1: An abstract view of movie selector.

and audio that are part of multimedia information sources.

Despite the similarities, on closer examination, certain clear distinctions can be made between information retrieval and the process of information filtering. Some of the typical distinguishing features are (Belkin & Croft 1992):

- Information retrieval usually implies more organization of the data (e.g., well defined storage and access mechanisms) so that user queries are made possible. On the other hand, information to be filtered tends to be of distributed nature and filtering systems typically deal with dynamic streams of information that are generated either by remote broadcasting sources (such as newswire services), or other direct sources (e-mail). Information filtering may also be needed if the incoming stream is generated by "intelligent agents" that search and retrieve remote heterogeneous databases.
- Information filtering is based on *profiles* that describe either individual or group preferences. Such profiles often represent long-term interests of the user. On the other hand, information retrieval from a database requires well defined user queries, and often, they reflect very short-term or instantaneous needs. Thus, depending on the degree of interactions, we can characterize information filtering as a "passive" (or, batch) process, while information retrieval is an "active" (or, on-line) process.

3 An Experiment on Movie Selection

3.1 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 and information retrieval. Namely, a user may be interested in watching a certain type of a movie at a particular time (which might be expressed in a query: "Tonight I want to see a good Steven Spielberg movie that I have not seen before"). At the same time the system might provide the user with periodic suggestions regarding new movie releases as they become available. Moreover, 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 (e.g., press releases, internet movie interest groups, movie critics, rating bodies, etc).

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 user equipment may consist of a set-top box and a multimedia terminal. To adapt the profile, a feedback loop that reflects the user's action on individual information items will be incorporated. The physical location of the profile is assumed to be part of the network. However, the profile can also be part of the set-top box.

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 redial 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[2] and a *feature-based* approach (Alspector & Karunanithi 1994; Karunanithi & Alspector 1996). 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. We present more details and an evaluation of these approaches in subsequent sections.

3.2 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

[2] The clique-based approach is a variant of the approach suggested by Will Hill of Bellcore. It has similarities to the "stereotype" of Rich (Rich 1983), the "communities" of Orwant (Orwant 1995), and "social filtering" as used by Maes (Maes 1994).

an information item (or a product) is generally known under the name of collaborative filtering (Goldberg et al. 1992). 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. Collaborative filtering has been applied, among others, to filtering of USENET group postings (Konstan et al., 1997), filtering of e-mail messages, and recommending interesting Web sites (Maes 1994). We are also aware of the existence of several internet sites where this method is used to provide movie recommendations[3].

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, C, which is a normalized dot product of the vectors of the ratings of two users. Thus for any two vectors **a** and **b**, the correlation similarity measure is defined as

$$C(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\sqrt{\mathbf{a} \cdot \mathbf{a}}\sqrt{\mathbf{b} \cdot \mathbf{b}}} \tag{1}$$

The values of C range from -1.0 to +1.0; that is, from perfect anti-correlation to perfect correlation. A value of zero corresponds to no correlation. Any other value has some predictive usefulness. Note that +1.0 is unrealistic because the identical user who rates the same movies a few weeks later will have a self correlation in the +0.8 to +0.9 range. 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}.

Given a target user, specifying proper values for C_{min} and S_{min} is not straightforward. 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

[3] http://www.filmfinder.com, http://rw.cinemax.com/critic, http://www.moviecritic.com, http://rw.cinemax.com/critic, http://www.moviefinder.com, http://movielens.umn.edu, http://vguide.sepia.com

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.

To implement this algorithm, we suggested how one could assign values to C_{min} and S_{min} and identify a clique. However, we did not make any specific recommendation as to what procedure should be used in step 3. We considered two approaches of aggregating the ratings of clique members. 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. The rationale behind computing the average is that a target user's taste may not be very different from that of his/her clique and that there is little differentiation within the set of correlations between a target user and its clique members.

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. Hence, clique members whose "taste" is better correlated with that of the target user have a larger influence on the subsequent rating predictions. The output of a correlation-ranking predictor is given by

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

where w_i is the correlation (weight) between the ith clique member and the target user. Although only these two approaches to aggregation of the clique ratings were considered, alternative (possibly nonlinear) strategies could certainly be proposed.

3.3 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. It is generally difficult to determine the relevant content of unstructured information sources. However, it has been shown that relatively simple approaches to this problem can be successful. For example, in filtering text documents it has been proposed to look for the occurrence of certain keywords (and their combinations), which proved to be quite effective.

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., parental guidance), expert critic ratings, movie category (e.g., drama), name of the director, leading actors/actresses, and awards received. For example, a particular user may have a strong inclination to see only movies that are rated G (general admittance), acted by Ben Kingsley, and belonging to the category Comedy or Drama. Thus, the feature-based approach exploits the bias of a user towards a set of important features of the movies.

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. For example, one could be interested in movies containing particular scenes (e.g., car chases) and containing certain type of dialog (e.g., avoiding harsh language). Some of the movie content could also be extracted via text-based methods (e.g., by analyzing the textual transcript), while others could be obtained through visual search of the video material.. In fact, a substantial amount of research is currently invested into visual/content-based querying of image and video data sources (Flickner et al. 1995, Aho et al. 1997), as more such information is being made available via the internet.

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 were a short version of the feature name is used later, the short version is provided in parentheses):

- *Category*: each movie can be described as belonging to one or more broad categories, such as comedy, drama, thriller, action or adventure. Based on the available movie data sources (detailed later), a set of 25 categories was considered.
- *MPAA rating* (MPAA): this feature represents the official rating given in the USA to movies approved for distribution. Every movie is assigned exactly one MPAA rating, indicating the age group for which the movie should be suitable. The six MPAA ratings are: G (general admittance), PG (parental guidance), PG-13 (parental guidance - age over 13), R (restricted admittance), X (adult viewers) and NC-17 (no children/adult viewers - age over 17).
- *Maltin rating* (Maltin): the ratings of professional reviewer Leonard Maltin. These ratings are originally on the scale from 0 to 4 (in a star-based system) and were transformed to the [0, 1] range for use in our experiments.
- *Academy Award* (AA): a movie could win the Academy Award (AA=1), be nominated for it (AA=0.5) or not be considered for the award at all (AA=0)
- *Length*: The length of a movie can have some influence on the target user. Here, for each movie in the database, its length (in minutes) was normalized with respect to the overall average movie length in the database.
- *Origin*: One could certainly have a preference (or dislike) for movies made in a particular country (e.g., made in Hollywood/USA). Considering that the user data used in our experiments were collected in the USA, three values of this feature were considered: 1. made in the USA (Origin=1), 2. made in the USA with foreign collaboration (Origin=0.5) and 3. foreign made (Origin=0).
- *Director*: People are known to have very definite opinions about movies by a particular director (and similarly about movies starring particular actors/actresses). The set of directors contributing to this feature clearly depends on the content of a particular database and is likely to grow as more movies are being added. Since the number of director "values" is very large, in order to simplify encoding for each user, 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.

The representation and format of these features (each taking an integer value in $\{0,1\}$ or a real value in $[0,1]$) is summarized in Table 1. The encodings in Table 1 are self-explanatory except for the features Category and MPAA, where we use a 1-of-N unary encoding[4].

Feature	Type of Encoding	No. of Input Units
Category	1-of-N unary	25
MPAA Ratings	1-of-N unary	6
Maltin Ratings	Real value between 0.0 and 1.0	1
Academy Award	Real value between 0.0 and 1.0	1
Length (minutes)	Real value normalized by the mean	1
Origin	Real value between 0.0 and 1.0	1
Director	Real value between 0.0 and 1.0 pre-processed for each user	1

Table 1: Encoding used for features.

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. In particular, addition of features relating to the leading actors/actresses and time of the release (e.g., 1930s as opposed to 1990s) could make the description more accurate. 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. The following sections provide a description of the architectures chosen.

3.3.1 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:

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

[4] Definition of 1-of-N unary encoding: Assume that we have N discrete values for the feature, and they are assigned a unique integer from 1 to N. If x is the value that is assigned an integer $1 \leq m \leq N$, then we construct a binary vector of length N such that there is only one "1" corresponding to x at the mth position and the rest are "0".

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.

3.3.2 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 preprocessing 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.

The strategy here is to create the MPAA and Category subnetworks first and then use thus processed features to design the top-level linear model. Learning in this case is performed in stages, instead of on all features at the same time as was the case in the fully linear approach. This network is a simplified form of a modular network in that the inputs for features Category and MPAA are first fed to separate (hidden) units rather than directly into the network. Also, since the MPAA subnetwork is designed according to a lookup-table approach (which is nonlinear), the overall model is no longer linear in its features. Additionally, when nonlinear sigmoidal functions (e.g., $y = 1/(1 + \exp(x))$) are applied to the output of the linear blocks in Figure 2, the model becomes equivalent with a neural network, which was applied to the problem of movie selection in our earlier work (Karunanithi & Alspector 1996).

3.3.3 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

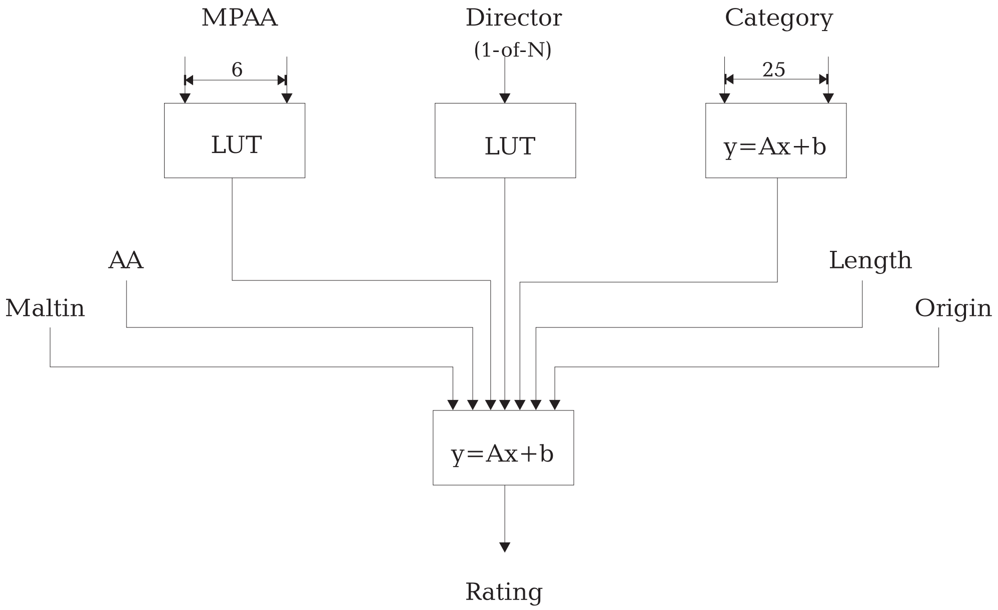
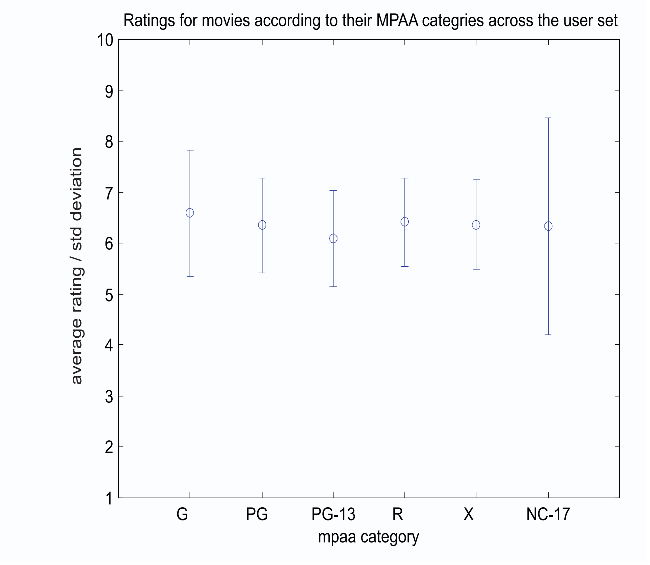


Figure 2: The linear network with MPAA and category feature groupings. The (y=Ax+b) blocks correspond to linear-network modules, whereas the (LUT) blocks represent lookup tables. The meaning of feature names (e.g., AA) is explained in Table 1. Wherever relevant, the original size/encoding of the features is indicated.

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 Figure 3) 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):

1. MPAA (low detail)
2. Category
3. Length, Origin, Maltin, AA



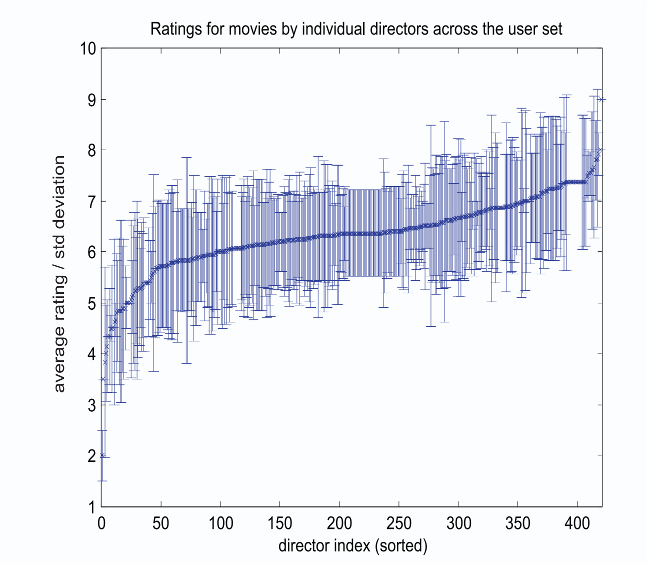


Figure 3: The average movie ratings (and their standard deviations) according to the MPAA categries (top) and movie directors (bottom). The bottom plot is ordered according to the average rating associated with movies by individual directors. The differentiation within the MPAA feature set appears to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

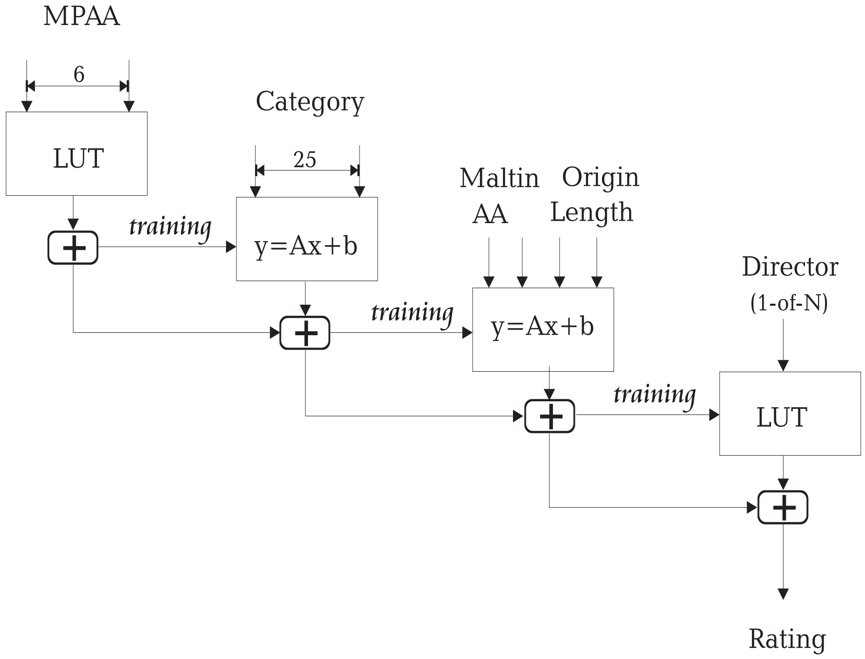


Figure 4: Architecture of the multiresolution network. The (y=Ax+b) blocks correspond to linear-network modules, whereas the (LUT) blocks correspond to lookup tables. The arrows marked as *training* indicate the signal flow used during network training. The meaning of feature names (e.g., AA) is explained in Table 1. Wherever relevant, the original size/encoding of the features is indicated.

4. 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. The learning proceeded from low-detail to high detail feature layers and the output of each layer was trained on the error between the output of the previous layer and the desired target. The network layers corresponding to the features grouped by the Category and (Length, Origin, Maltin, AA) were modeled by linear networks, whereas the MPAA and director features led to subnetworks of the lookup-table type.

3.3.4 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)

(Breiman et al., 1984) 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. The regression type of CART considered here represents a binary tree, which provides a piecewise-constant approximation of the target function, where each region of constant value is delimited by hyper-planes perpendicular to the system axes (a histogram-like function). Each internal node of the tree corresponds to a "split" on one of the variables (representing movie features in our case), with the split value selected so as to minimize the error of fit. Given an input datum, it is "dropped" down a tree and after following a series of splits at the internal tree nodes, the point is finally assigned to (exactly) one of the terminal nodes (i.e., tree leaves). The process of designing a tree starts with all (training) data points belonging to a single tree node (i.e., the root). Subsequently, each terminal node is considered in turn, where for all variables, the values of each variable possessed by points falling into that node are examined as potential candidates for a split. Each such split leads to a reduction of the mean-squared error for points falling into that node and the best overall split is selected. The node splitting processes can be continued until every terminal point contains only few data points (e.g., just one). However, in order to avoid overfitting, a form of regularization is used, where the error-of-fit is weighed against a penalty term given by the size of the tree. Thus, the regularized error E_R is given by

$$E_R = E + \alpha \cdot \left|\widetilde{T}\right| \tag{5}$$

where E is the resubstitution error (i.e., the squared Euclidean distance between the training set target values and their CART approximations), α represents a regularization parameter, and $\left|\widetilde{T}\right|$ denotes the size of the tree defined as the number of its terminal nodes. A highly overfitted tree is originally grown on the training data and subsequently pruned into a sequence of trees of decreasing size, each being optimal for a range of the regularization parameter (see Breiman et al. 1984 for details). The overall optimal regularization parameter (Girosi et al. 1995) α is selected by cross validation and is used to choose the final CART tree.

An example of a CART tree grown on the user movie data is shown in Figure 5.

3.4 A CART Network with Bagging

Bootstrap aggregation (bagging) (Breiman, 1996) represents a technique where the variance component of the generalization error of a given network can be reduced by aggregating several variants of the network designed using slightly different versions of the same input data. In particular, given an input data set of size N, the data are used to create several different sets of the same size N via sampling with repetitions from the original set (i.e., bootstrap sampling).

Each of these sets is, in turn, used to create a network of the given type, where the overall output of such a system is defined as an average of the individual network outputs.

Bagging is effective for "unstable" networks that are inherently sensitive to initial conditions and small changes of the data set (e.g., neural networks or CART) and does not offer any advantages for networks that are resistant to the variations of the input data, such as the linear networks. In fact, application of bagging to stable (e.g., linear) models can degrade their performance.

4 Results

4.1 Data Collection

The feature database for the experiment was initially populated from the Microsoft Cinemania[5] CD-ROM for 1548 movies in Will Hill's survey (Hill et al. 1995), 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 0 to 10, with 0 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). 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 2 shows the numbers of movies rated by users from the target groups (middle row), as well as the numbers of movies rated for which all features were available (bottom row). 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

[5] Cinemania is a registered trademark of Microsoft Inc.

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

Table 2: The numbers of movies rated (second row) by individual target users shown together with the respective numbers of movies for which all features were actually available (third row).

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.

4.2 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.

4.3 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 3 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

User ID	Clique	Clique with ranking	Clique (reduced)	Clique (reduced) with 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 3: 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.

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.

4.4 The Feature-Based Approach

4.4.1 Analysis of the CART Networks

A typical single CART tree grown on user data (user 39 in this case) is shown in Figure 5. 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 6. 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.

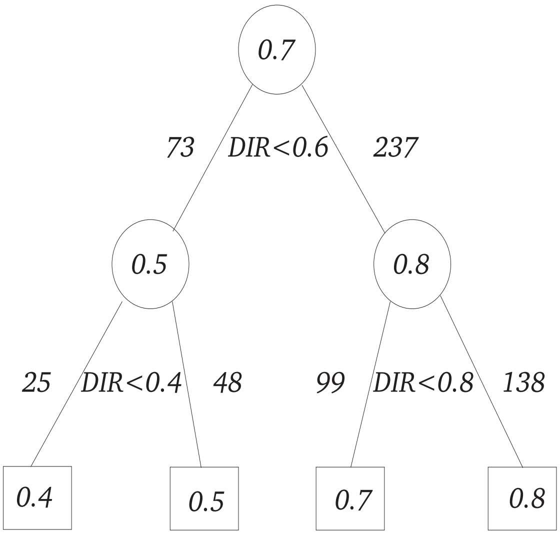


Figure 5: Example of a tree created for user U39; the DIR label corresponds to the Director variable (representing the average rating given to movies by a particular director by the target user); a number within a circle, or a rectangle, corresponds to the average rating of movies falling into the node; the numbers beside the connecting lines represent the numbers of training points following a particular split.

4.4.2 Performance Comparison

Table 4 shows the combined results of our study. The results are presented in terms of the correlation coefficient (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 4 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 4) is also added to assess to what extent user-dependent modelling 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 user U145) than the ones obtained with the feature-based modelling, although most of the feature-based networks provided an improvement of the Maltin rating (a movie rating expert). 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 subnetworks 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. As far as CART is concerned, 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 methods). However, these results suggest that the selected set of features (apart from Director) is not very informative and certain additions should be made to make the feature approach truly competitive with the clique method. Incorporating the leading actor/actress features would probably boost the performance to a certain extent, although these features could suffer for the same reason as the Director feature — that is, relatively few movies in the data set of any given user will share the same leading actor/actress, which makes generalization based on this feature difficult. To avoid these problems, clustering methods might be applied to group individual values of these features (e.g., directors), which would make the rating prediction less sensitive to small changes in the input data.

Figures 7–11 illustrate the predictive performance of the clique and feature-based models for each target user, where the results shown correspond to the clique method with correlation-based ranking and the linear model with MPAA and Category feature grouping. For better visualization, in each case the movies were sorted according to the rating provided by the target user. As can be seen, both the clique and the feature-based models show a significant spread about the desired target ratings. However, the ratings due to the feature-based model tend to be clustered more about their average, which leads to larger errors for movies which are rated very high or very low by the target user. The averaged root mean squared error for the clique and feature models used in Figures 7–11 is 1.62 and 1.49, respectively. Notice that although the clique models show better correlation with the user ratings, in some cases they systematically over rate (for users U64 and U124) or under rate (for users U21, U39, U111 and U145) in their movie-rating predictions, which leads to a higher bias of the estimates and results in higher mean squared errors. No such bias can be observed in

User	Maltin	Linear	Linear w.	Multi-	CART	CART w.	Clique w.
ID	rating		grouping	resolution		bagging	ranking
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 4: 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.

feature-based models.

5 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 (Karunanithi & Alspector 1996). 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. Results obtained in this study show that clique-based methods may have an advantage as far as capturing the extreme rating preferences of users (although clique-based predictions may be noticeably biased), as the feature-based models resulted in poorer predictions for movies that the target users considered very good or very bad. This, however, can be also attributed to the purposefully small set of features used in this study.

Our experiments with several feature-based networks suggest that, for the set of features chosen, a good model can be built on the foundation of a linear-type architecture. At the same time, sensitivity of the feature-based models to features such as movie director has been revealed. Features such as movie director or the lead actor/actress can have a large influence on a movie rating. However, they are difficult to generalize. The results obtained in this study suggest that an effective way of incorporating such features into a network model must

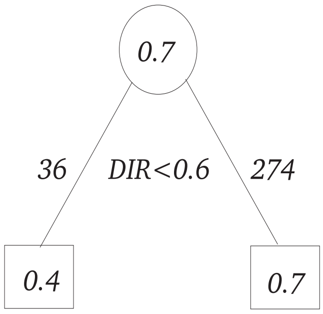


Figure 6: Example of a tree created for user U39 with the modified encoding of the Director feature (compare with Figure 5). As the variability of this feature is significantly decreased a much smaller tree results.

overcome this sensitivity, perhaps through clustering methods, where groups of directors/actors associated with movies of similar rating (for a given target user or a clique of users) are grouped together.

Each of the approaches investigated has its own strengths and limitations. First, in terms of feedback, both approaches need active participation from the user. Unless the interface is simple and provides easy-to-use features, it may not be easy to convince the user to provide useful feedback. Second, the clique-based approach cannot be used for new movies (although, due to the internet connectivity, the data gathering process can be quite fast) and for users without a clique. So, the feature-based approach may provide an advantage if a new movie needs to be selectively targeted for the customers. Third, identifying a clique may pose problems in terms of sharing information with other users and related privacy issues. Fourth, the feature-based approach requires a careful selection, extraction, and representation of features. Considering these arguments, we believe that an effective movie-recommendation system should combine both approaches to maximize its performance. One possible way would be to use a mixture-of-experts approach (Jordan & Jacobs, 1994), where the results provided by individual rating modules would be combined though a weighted average, with the weighting coefficients being adaptive and depend on the amount of data available to the clique.

Some of the models presented here are in the process of being incorporated into our Movie Database and Recommendation System (MARS) internet site[6], which we intend to use for gathering more data and user feedback. In the future we aim to utilize richer feature sets (as well as investigate other modeling strategies) and combine the clique and feature-based approaches into a single hybrid system.

Acknowledgments

[6] http://lunar.uccs.edu/movies

We are indebted to Will Hill of Bellcore for allowing us to use his movie list and ratings by internet users. We are also thankful to Thomas Judd of Bellcore for populating the database with features. We would also like to thank Brian Crowder and Ritchie West for their help with designing the movie database.

References

Aho, A., Chang, S. F., McKeown, K., Radev, D., Smith, J. and Zaman, K.: 1997, Columbia digital news system: An environment for briefing and search over multimedia information, *Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries (IEEE ADL '97)*, IEEE Computer Society Press, Los Alamitos, CA, pp. 82–94.

Alspector, J. and Karunanithi, N.: 1994, Smart interfaces for the NII, *Proceedings of Focused Program Development Workshop on Networking, Telecommunications and Information Technology*, pp. 362–367.

Balabanivić, M. and Shoham, Y.: 1997, Content-based collaborative recommendation, *Communications of the ACM* **40**(3), 66–72.

Belkin, N. J. and Croft, W. B.: 1992, Information filtering and information retrieval: Two sides of the same coin?, *Communications of the ACM* **35**(12), 29–38.

Breiman, L.: 1996, Bagging predictors, *Machine Learning* **24**, 123–140.

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

Danzig, K. O. P. and Li, S.: 1993, Internet resource discovery services, *IEEE Computer* **26**(9), 8–22.

Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Qian, H., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petcovic, D., Steele, D. and Yanker, P.: 1995, Query by image and video content: the QBIC system., *Computer* **28**(9), 23–32.

Girosi, F., Jones, M. and Poggio, T.: 1995, Regularization theory and neural network architectures, *Neural Computation* **7**, 219–269.

Goldberg, D., Nichols, D., Oki, B. M. and Terry, D.: 1992, Using collaborative filtering to weave the information tapestry, *Communications of the ACM* **35**(12), 61–70.

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

Jordan, M. I. and Jacobs, R. A.: 1994, Hierarchical mixtures of experts and the em algorithm, *Neural Computation* **6**, 181–214.

Karunanithi, N. and Alspector, J.: 1996, Feature-based and clique-based user models for movie selection, *Proceedings of the Fifth International Conference on User Modeling*, User Modeling, Inc., Publishers, Kailua-Kona, HI, pp. 29–34.

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R. and Riedl, J.: 1997, GroupLens: applying collaborative filtering to Usenet news, *Communications of the ACM* **40**(3), 77–87.

Maes, P.: 1994, Agents that reduce work and information overload, *Communications of the ACM* **37**(7), 30–40.

Orwant, J.: 1995, Heterogeneous learning in the doppelganger user modeling system, *User Modeling and User-Adapted Interaction* **4**(2), 107–130.

Rich, E.: 1983, Users are individuals: Individualizing user models, *Int'l Journal of Man-Machine Studies* **18**, 199–214.

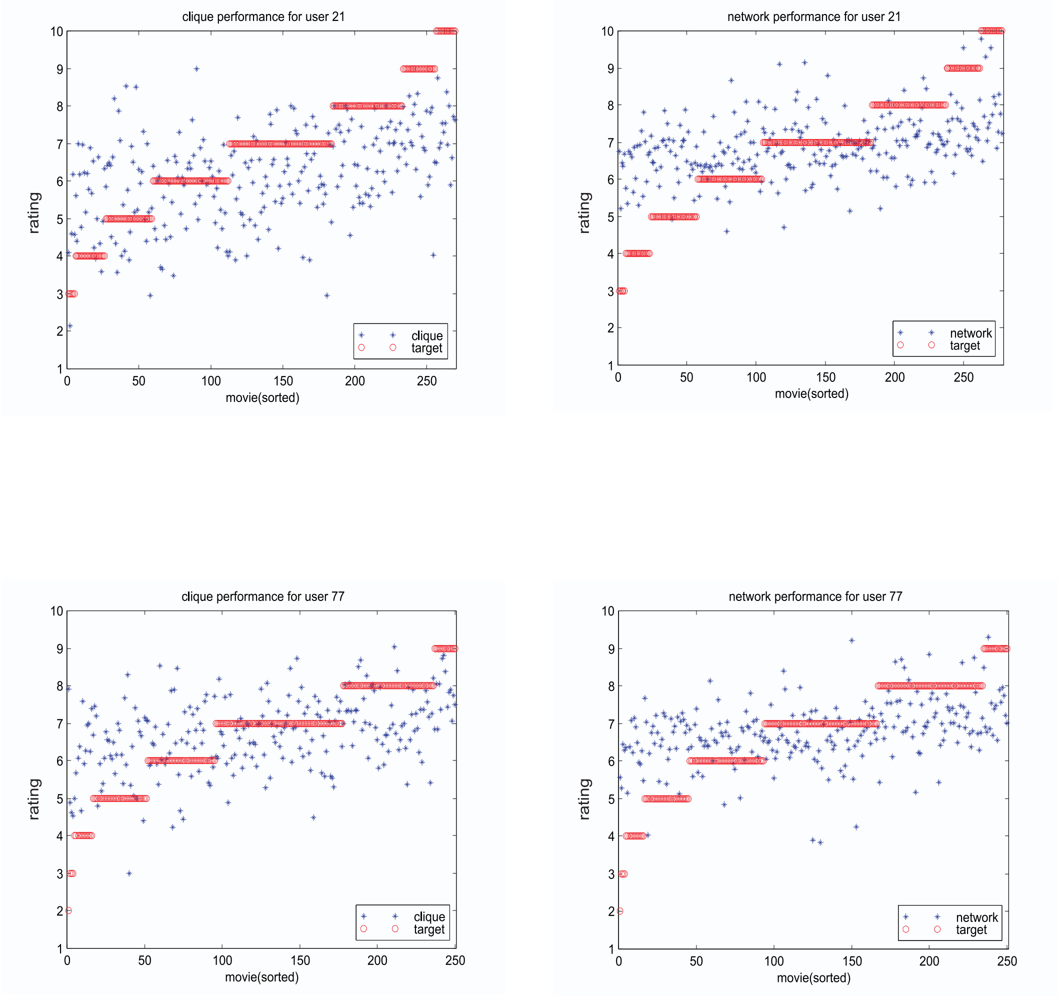


Figure 7: Rating prediction performance of the clique method with correlation ranking (left) and the linear model with MPAA and Category feature grouping (right) for user U21 (top) and user U77 (bottom). For better visualization, in each case the movies have been ordered according to their target rating. Ideally, for each value of the user target rating (represented by a horizontal line of circles), the predictions provided by the model (represented by a “cloud” of pluses) should closely agree with the user ratings (i.e., ideally, there should be a matching a horizontal line of pluses in each case).

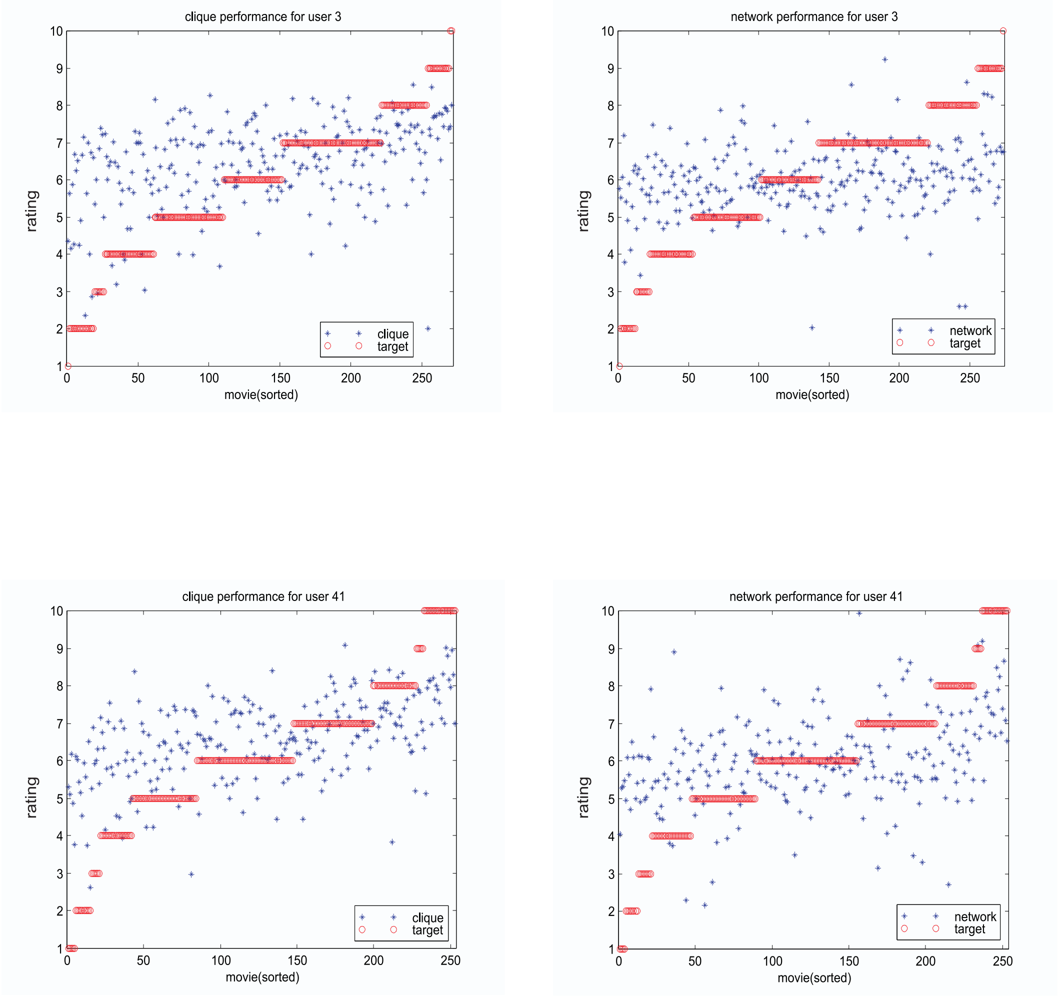


Figure 8: Rating prediction performance of the clique method with correlation ranking (left) and the linear model with MPAA and Category feature grouping (right) for user U3 (top) and user U41 (bottom). For better visualization, in each case the movies have been ordered according to their target rating. Ideally, for each value of the user target rating (represented by a horizontal line of circles), the predictions provided by the model (represented by a "cloud" of pluses) should closely agree with the user ratings (i.e., ideally, there should be a matching a horizontal line of pluses in each case).

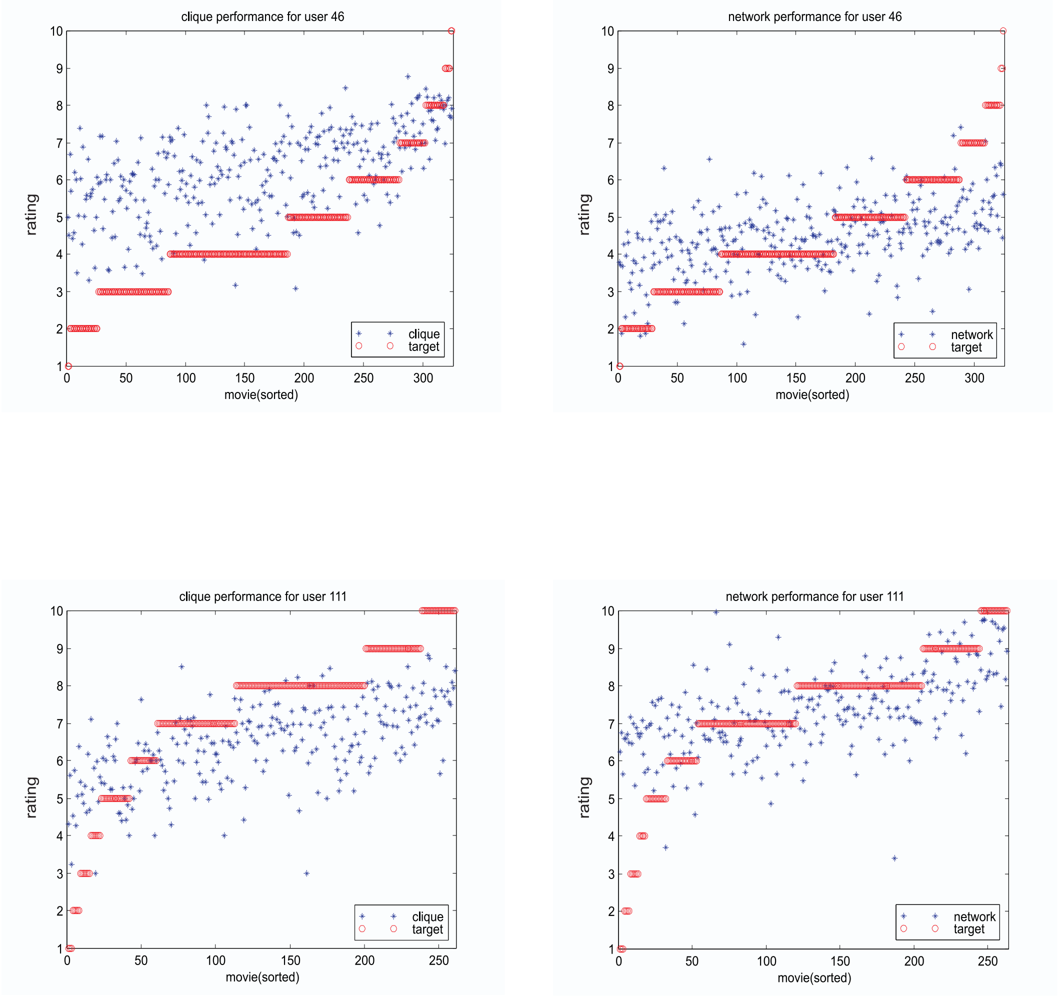


Figure 9: Rating prediction performance of the clique method with correlation ranking (left) and the linear model with MPAA and Category feature grouping (right) for user U46 (top) and user U111 (bottom). For better visualization, in each case the movies have been ordered according to their target rating. Ideally, for each value of the user target rating (represented by a horizontal line of circles), the predictions provided by the model (represented by a "cloud" of pluses) should closely agree with the user ratings (i.e., ideally, there should be a matching a horizontal line of pluses in each case).

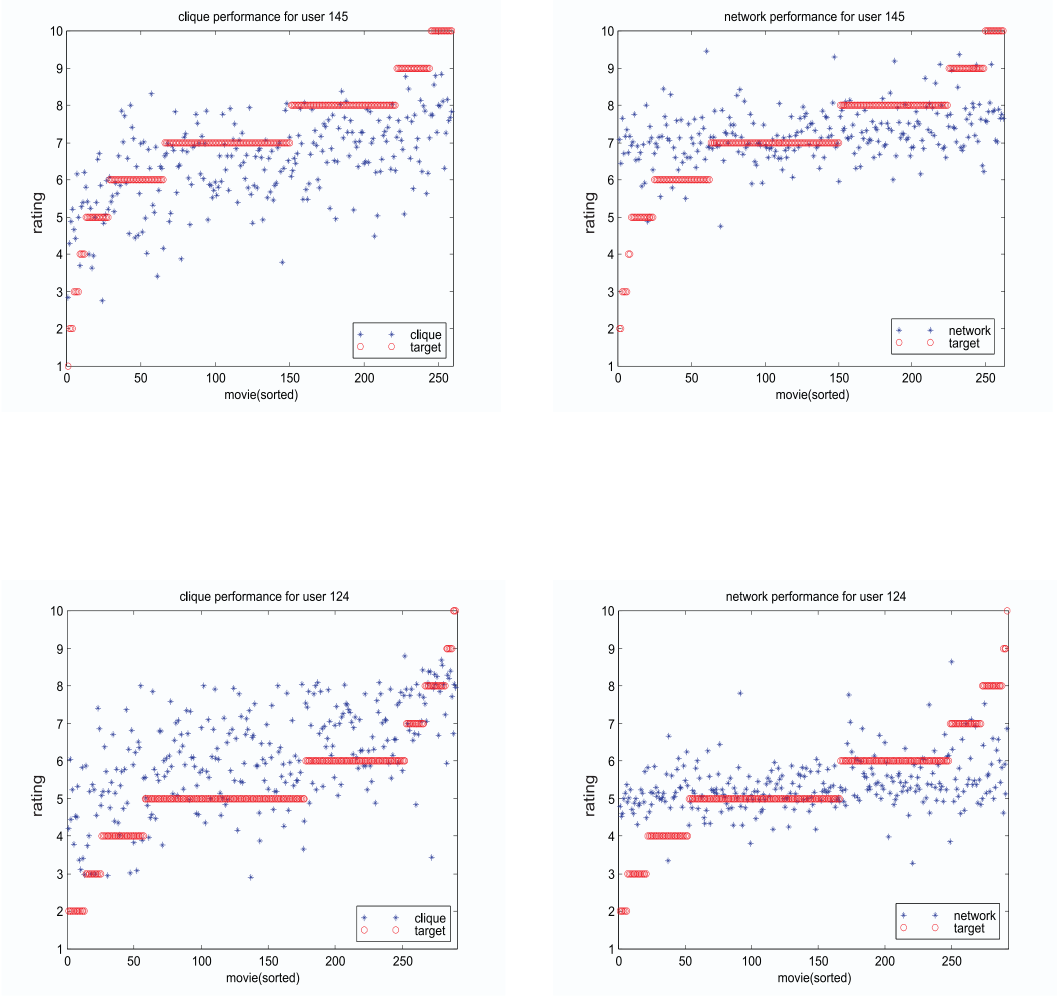


Figure 10: Rating prediction performance of the clique method with correlation ranking (left) and the linear model with MPAA and Category feature grouping (right) for user U145 (top) and user U124 (bottom). For better visualization, in each case the movies have been ordered according to their target rating. Ideally, for each value of the user target rating (represented by a horizontal line of circles), the predictions provided by the model (represented by a “cloud” of pluses) should closely agree with the user ratings (i.e., ideally, there should be a matching a horizontal line of pluses in each case).

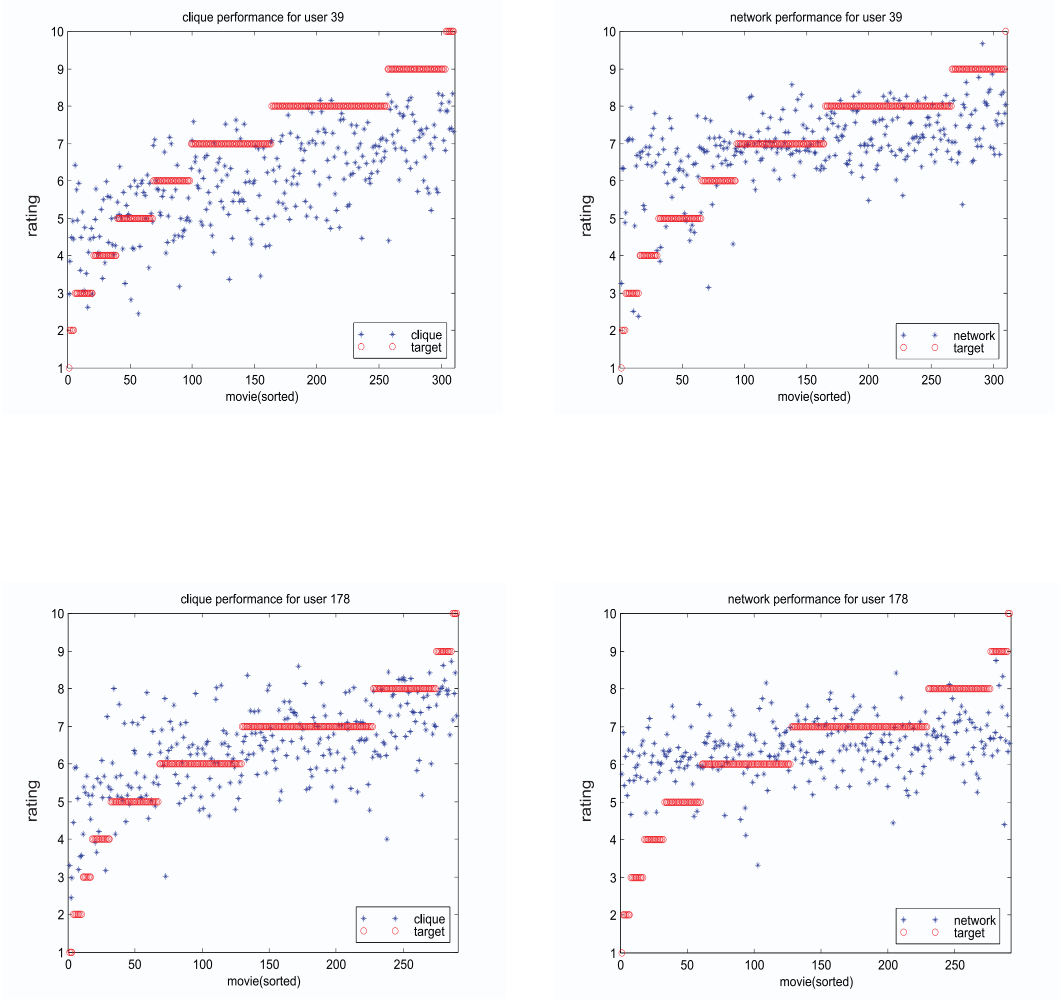


Figure 11: Rating prediction performance of the clique method with correlation ranking (left) and the linear model with MPAA and Category feature grouping (right) for user U39 (top) and user U178 (bottom). For better visualization, in each case the movies have been ordered according to their target rating. Ideally, for each value of the user target rating (represented by a horizontal line of circles), the predictions provided by the model (represented by a “cloud” of pluses) should closely agree with the user ratings (i.e., ideally, there should be a matching a horizontal line of pluses in each case).