The Decision-Theoretic Video Advisor*

Hien Nguyen and Peter Haddawy
Decision Systems and Artificial Intelligence Lab (DSAIL)
Dept. of Electrical Engineering and Computer Science
University of Wisconsin-Milwaukee
Milwaukee, WI 53201
{nguyen,haddawy}@cs.uwm.edu

Abstract

We describe ongoing work toward development of a decision-theoretic agent to help users choose videos based on their preferences. The DIVA (Decision-Theoretic Interactive Video Advisor) system elicits user preferences using a case-based technique. Hard constraints are used to permit the user to communicate temporary deviations from his basic preferences. If the user is not happy with the system's recommendations, he can provide feedback, which is used to modify the represented preferences and generate a new set of recommendations. We describe the fundamental algorithms, the implementation, and some results from some initial experimentation.

Introduction

We are interested in exploring the issues involved in providing users with preference-based access to information. Preference-based search is required, for example, when a user wishes to search an online catalog for a desired item or items but is unfamiliar with the domain or with the exact contents of the catalog. Take the example of searching an online video catalog for a video to rent. In such a search, a user is looking for one or a few movies to rent. A standard data base query would typically return a large set of movies or possibly an empty set of movies, neither of which satisfies the user's objective. The binary distinction between items that satisfy a description and those that do not is too coarse. As an alternative, we describe a decision-theoretic approach to searching large databases. We have chosen to work within the domain of video selection for several reasons. First, it is a domain that can be understood by anyone. So people can easily evaluate the performance of the system. Second, several large movie databases with rich sets of attributes are available. Finally, movie selection has been extensively studied using other techniques. This will facilitate comparing our approach to previous approaches.

This work was partially supported by NSF grant IRI-9509165.

The predominant approach to preference-based selection of movies is collaborative filtering (MovieLens, MovieCritics, FireFly [1]). This approach makes recommendations to a user based on the opinions of other users in the collaborative group who have tastes similar to that user. The approach requires users to rate movies on some numeric scale. The more films users rank and the more careful they are about their ranking, the better the system performs. This approach has the drawback that it considers for recommendation only those movies that have been ranked by at least one user in the collaborative group. Furthermore, the approach does not make direct use of attributes characterizing the movies, e.g. genre, running time, casting. The work on collaborative filtering is not cast in the framework of Decision Theory and no theoretical framework or justification are provided for the similarity measures used in matching users' tastes. In contrast, our approach uses utility theory as the underlying framework for representing user preferences and and makes heavy use of attributes for preference elicitation. We combine attribute-based elicitation of user preferences with matching of a user's preferences against those of other users of the system.

In this paper, we describe ongoing work toward development of a Decision-Theoretic Interactive Video Advisor (DIVA). The rest of the paper is organized as follows. Section 2 presents the techniques used for representing and eliciting preferences. Section 3 provides an overview of the system and describes the implementation. Future research is discussed in Section 4.

Representation and Elicitation Techniques

Preference Representation

We expect that a user will have a basic taste in movies that is fairly stable over time, as well as preferences

[1] Available at http://movielens.umn.edu, http://www.moviecritics.com, and http://www.firefly.net.

that change between uses of the system, i.e., what the user is interested in watching at this particular time. For example, someone may have a general preference for dramas with good cinematography, but they may be more in the mood for an action movie this evening. An advisory system should be flexible enough to take into account the temporary preference for action movies over dramas but should also use the preference for good cinematography to rank the action movies. We handle this by representing the person's long-term preferences with a value function and the short-term preferences with hard constraints. We combine the two by first using the hard constraints as a filter on the data base of movies and then using the value function to rank the remaining movies. By doing this we obtain the desired system behavior.

Decision theory provides a rich theoretical framework for representing preferences (R. L. Keeyney 1976). According to decision theory, a user's preferences over a space of outcomes can be represented by a real-valued function over the space. When no uncertainty is involved, as is the case in the video domain, the function is called a value function. The outcome space is typically defined in terms of a set of attributes, with an outcome being a complete assignment of values to all the attributes. A value function is typically specified in terms of component functions for each of the attributes. Unfortunately, traditional techniques for eliciting value functions are typically time consuming and tedious. To simplify the task, one is forced to make assumptions concerning structure of the value function such as preferential independence over the attributes or additive utility model. One of the difficulties we meet in the movie selection domain is that we can not assume preferential independence among all attributes. Take the director, casting and genre attributes for example. A user may like James Cameron's dramas and romances but may dislike his action movies. The same thing happens with the casting: a user may like Meg Ryan in comedies and romances but not like her performances in an action movies. In addition, in low-stakes decision making problems such as movie selection, it is unreasonable to expect a user to spend a large amount of time expressing his preferences to an advisory system. These two difficulties urge us to explore techniques that accelerate the preference elicitation process without making restrictive assumptions concerning the form of the underlying value function.

Existing movie databases such as The Internet Movie Database [2] contain as many as 40 attributes to describe each movie. As that is too many to work with, we decided to narrow down the set of attributes and classify them into groups including preference attributes and constraint attributes. The preference attributes are ones over which the user can express his preferences (e.g casting, director, genre). They are involved directly in the preference elicitation process. The constraint attributes can be used to focus the search if the user has some specific information (e.g sound track, movie title, character name) about the kind of movie he is looking for. We chose the attributes director, casting, genre, star rating and running time as preference attributes. Among these five chosen attributes, we can see that star rating and running time are preferentially independent of the other attributes. Unfortunately, we can not say any one of the remaining three attributes is preferentially independent of the other two. Based these observations, we can represent the value function as follows.

$$v(cast, director, genre, rating, time) = \lambda_1 u_1(time) + \lambda_2 u_2(rating) + \lambda_3 u_3(casting, director, genre),$$

where λ_i represents the importance of the attribute i to the user, and u_i is a subvalue function for the attribute i.

Case-Based Preference Elicitation

To elicit the value function representing the user's long-term preferences, we use the case-based technique described in (Ha & Haddawy 1998). The idea of this technique is based on the observation that people tend to form clusters according to their preferences or tastes. We maintain a group of users with their preferences over movies partially or completely specified. When a new user comes in, his preference structure will be determined partially and then matched against the preference structures of the existing group of users. We seek the user from the existing group who has a preference structure that is closest to the partially elicited preference structure of the new user. The retrieved closest matching preference structure is then used to supplement the partially elicited user preferences.

The remaining problem to be solved is how to perform the initial partial elicitation of the user's preferences. Recall that the value function is composed of three subvalue functions: two over one attributes and one over three attributes. We allow the user to select from one of a few prototype subvalue functions over star rating and over running time. We make some assumptions concerning the forms of the subvalue functions. For example, we assume that value functino over running time linearly increases over some range of values, is then constant over some interval, and linearly

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

descreases as time increases further. Eliciting the subvalue function for casting, director and genre is much more difficult since that function has no regularity. We use the inductive learning program C5.0 [3] to elicit the function. This is done by assuming the function is binary valued and asking the user to provide a list of movies he particularly likes and a list of movies he particularly dislikes. We assume that if he likes a movie, he would also like another movie with the same (casting, director, genre) combination. These lists are used as labeled training instance to C5.0 in order to come up with the subvalue function.

Ha & Haddawy (Ha & Haddawy 1998) define a probabilistic distance measure between preference structures as the probability that two randomly chosen elements are ranked differently by the two preference structures.

$$\begin{aligned}\delta_P(\prec_1, \prec_2) &= \Pr(\prec_1 \,\&\, \prec_2 \text{ rank } s_j \text{ and } s_k \text{ differently}) \\ &= \frac{2}{n(n-1)} \sum_{1 \le j < k \le n} c_{\prec_1, \prec_2}(s_j, s_k),\end{aligned}$$

in which

$$c_{\prec_1, \prec_2}(a,b) = \begin{cases} 1 & \text{if } (a \preceq_1 b \wedge b \prec_2 a) \vee \\ & \quad (a \prec_1 b \wedge b \preceq_2 a) \vee \\ & \quad (a \preceq_2 b \wedge b \prec_1 a) \vee \\ & \quad (a \prec_2 b \wedge b \preceq_1 a) \\ 0 & \text{otherwise.} \end{cases}$$

$\prec_1$ and $\prec_2$ are two weak orders on the finite set of outcomes $\Omega = \{s_1, s_2, ...s_n\}$

Since our objective is to use retrieved cases to supplement partially elicited preferences, we need a distance measure over partially specified preference structures. A partial preference structure can be thought of as a partial order. A partial order can, in turn, be thought of as a specification of a set of linear extensions - the complete preference structures consistent with it. Ha & Haddawy (Ha & Haddawy 1998) define the distance between two partial preference orders as the probability that two randomly chosen elements are ranked differently by two randomly chosen linear extensions.

Computing this probability is closely related to the problem of computing the number of linear extensions of a finite partial order. That problem is known to be #P-complete in (Brightwell & Winkler 1991). We use the approximation algorithm of (R.Bubley & M.Dyer 1998) which reduces the problem of counting linear extensions to one of sampling and uses a Markov chain algorithm to do the sampling.

[3] http://www.rulequest.com

Determining the User's Initial Preferences

As mentioned in Section 1, we use the inductive learning algorithm C5.0 to determine the user's initial preference structure. When a new user comes in, he is asked to create an account and list movies that he particularly likes and dislikes. We call these lists "like list" and "dislike list" respectively. By creating these two lists, we implicitly use the assumption that if he likes a movie, he would also like another movie with the same (casting, director, genre) combination. The user also be asked to express his preference in term of running time and movie star rating. We make what we consider to be reasonable assumptions concerning the structure of these two subvalue functions. We assume that preferences over star rating are monotonic and that preferences over running time are monotonically increasing, then constant, and then monotonically decreasing.

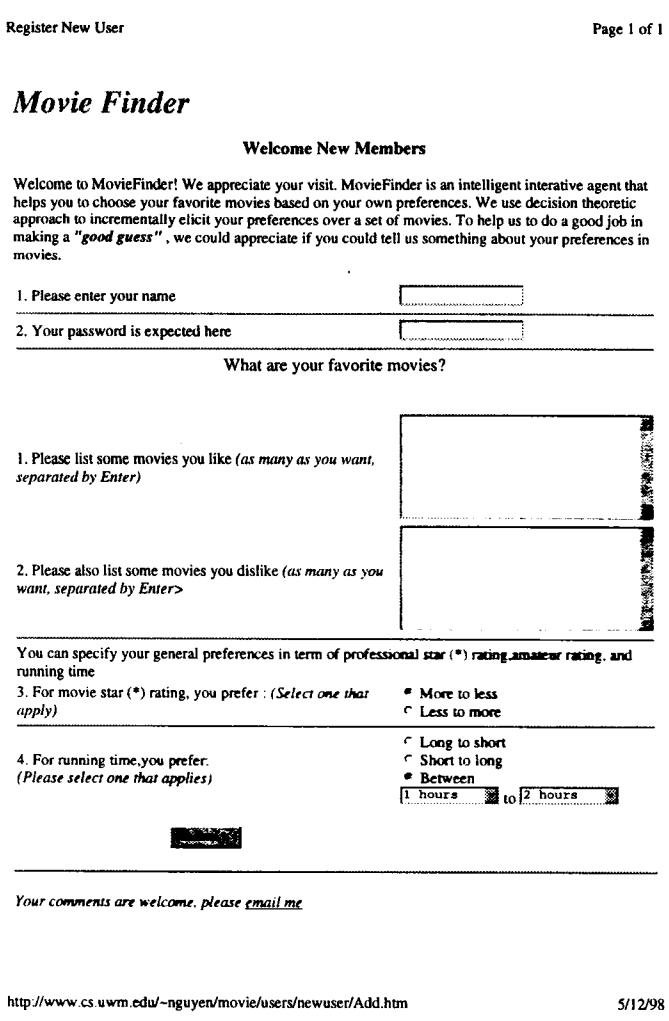
Empirical results

To perform an initial evaluation of our approach, we created a small user database. Ten graduate students were asked to rank fifty movies on a scale from 0 to 10 in increments of 0.1. Higher values represent more preferred movies. The rankings were used as cases to be retrieved.

We evaluated the matching algorithm by seeing whether a partial preference structure for a user would match that user's complete preference structure in a case base that contained other users as well. We did this by selecting one of the preference structures from the case base and then using it to simulate partial elicitation from a new user. First we take some movies he likes and dislike for the "like" and "dislike" lists respectively. For example, we used the seventh user in the case base. His initial partial preference structure is shown in Table 1. By looking at the list of movies he likes, we can see the fairy obvious pattern of preference. He appears to like action moviees and dislikes comedies. A part of the completely preference structure of all the user is shown in the following table:

With the number of steps in the sampling algorithm set at 100, we found that the distance measure is fairy stable, i.e., the seventh user is always matched closest with himself in the case base.

In order to further evaluate the matching algorithm, we added to our case base some fictional users who liked a particular genre of movie over all other genres. We then entered preferences for a new user who likes comedies or action movies and matched against the appropriate user in the case base.



Register New User Page 1 of 1

Movie Finder

Welcome New Members

Welcome to MovieFinder! We appreciate your visit. MovieFinder is an intelligent interative agent that helps you to choose your favorite movies based on your own preferences. We use decision theoretic approach to incrementally elicit your preferences over a set of movies. To help us to do a good job in making a *"good guess"*, we could appreciate if you could tell us something about your preferences in movies.

1. Please enter your name
2. Your password is expected here

What are your favorite movies?

1. Please list some movies you like *(as many as you want, separated by Enter)*
2. Please also list some movies you dislike *(as many as you want, separated by Enter>*

You can specify your general preferences in term of professional star (*) rating,amateur rating, and running time

3. For movie star (*) rating, you prefer : *(Select one that apply)* * More to less ○ Less to more
4. For running time,you prefer: *(Please select one that applies)* ○ Long to short ○ Short to long * Between 1 hours to 2 hours

Your comments are welcome, please email me

http://www.cs.uwm.edu/~nguyen/movie/users/newuser/Add.htm 5/12/98

Figure 1: A screenshot of the registration window for a new user.

	Movies
Liked Movies	Apolo 13 Twister Independence Day Cop Land The Relic
Disliked Movies	The First Wives Club My Best Friend Wedding

Table 1: Initial list of movies liked and disliked by a user.

Discussion and Future Research

This paper describes the first version of the DIVA system. Several difficult technical problems remain to be solved. We currently use the partially elicited preferences to simply retrieve a preference structure from the case base. But that structure will typically have some conflicts with the partially elicited preferences. We would like to be able to merge the two, retaining all the partially elicited preferences. Our distance measure can be used to define an approprite concept of a merged preference structure, but how to compute it is an open question. We are investigating how the techniques for relevance feedback from information retrieval (G. Salton 1983) can be adapted to our framework. If none of the recommended movies is what the user had in mind, we would like to provide him with the ability to indicate which movies out of the recommended list are the closest to what he was looking for and to use that information to modify the preference structure. We are also looking at ways of dealing with computational complexity for case bases with large numbers of users and data bases with large numbers of movies. If our case base has a large number of preference structures, we may want to hierarchically cluster them in order to speed up the search. A real data base of movies will typically contain on the order of 20,000 movies. Computing distances between partial orders over 20,000 elements can be extremely time consuming. One way to speed the computation is to approximate the distance by computing it only over some fraction of the movies. For example, we may choose to compare only the top-ranked n movies.

References

Brightwell, G., and Winkler, P. 1991. Counting linear extensions is #p-complete. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, 175–181.

G. Salton, M. J. M. 1983. *Introduction to Modern Information Retrieval.* McGraw-Hill Book Company.

Ha, V., and Haddawy, P. 1998. Toward case-based preference elicitation: Similarity measures on preference structures. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence.* Forthcoming.

R. L. Keeyney, H. R. 1976. *Decision with Multiple Objectives: Preferences and Value Tradeoffs.* John Wiley and Sons.

R.Bubley, and M.Dyer. 1998. Faster random generation of linear extensions. In *Proceedings of the Ninth Annual ACM-SIAM Symposium of Discrete Algorithms*, 175–186.