A Multi-Agent TV Recommender

Kaushal Kurapati, Srinivas Gutta, David Schaffer, Jacquelyn Martino, John Zimmerman

Adaptive Systems Department
Philips Research Briarcliff
345 Scarborough Rd.
Briarcliff Manor, NY 10510, USA
Phone: 1-914-945-6231
E-mail: kaushal.kurapati@philips.com

Abstract.

Personal Television is here via the advent of a new class of devices called personal video recorders (PVRs). These recorders change the user task from (a) selecting a specific channel to watch from the 100+ available channels to (b) finding something "good" to record from the 10,000+ shows broadcast each week. Recommender systems, such as the one described in this paper, will help track users' preferences and aid users in choosing shows to record. In this paper we advance a multi-agent TV recommender system that encapsulates three user information streams--implicit view history, explicit preferences, and feedback information on specific shows--into adaptive agents and generates program recommendations for a TV viewer. We have tested the system in various agent combinations with real users drawn from a wide variety of living conditions. The combination of implicit and explicit agents seems to work best in our framework.

Keywords: Multi-agent TV recommender system, user profiling, machine learning, user testing.

Introduction

The advent of a new class of consumer electronic (CE) devices, called personal video recorders (PVRs), from companies such as TiVo© and Replay TV©, has begun to change the way people watch TV. PVRs are tapeless, hard disk equipped devices that let TV viewers record shows via an attractive screen based user-interface. These devices put users in control. Now every time a user sits down to watch TV there is always something "good" to watch on the disk. Users are no longer slaves to the broadcaster's schedule. In short, it is about Personal Television--getting what you want to watch, when you want to watch it.

PVRs can easily be programmed to record all of the programs users want; however, how can users find what they want? Previously users were faced with an increasing number of choices from cable and satellite TV. They were often choosing from 50 to 300+ channels. The arrival of the PVR changes this equation. Instead of choosing a program to watch now from the current 300 shows that are being broadcast, users need to select something "good" to record from the tens of thousands of shows broadcast each week so that they can watch them when they choose.

The advent and meteoric rise in popularity of the Internet has motivated many people to go online and has made them conversant with the personalization tools available on the web. As the trend towards media convergence in the home gains momentum, user expectations for tools that help them find, personalize, and organize content will rise. For a PVR to provide an enriched TV experience to the user, personalization is the key. PVRs need to be equipped with sophisticated recommender systems that track and recognize user preferences and help them select good content to fill the hard disk. This paper presents our research on a multi-agent system for finding, recommending, and recording TV programs.

Relevant Work

Much work has been done in the area of recommender systems and they have been applied to a wide range of disciplines. Many systems have been built in recent years to help users deal with large quantities of information coming from various sources: e-mail (Maxims [Lashkari, et. al.]), usenet news (NewT [Sheth and Maes]), the web (Letizia [Lieberman], Syskill & Webert [Pazzani, et. al.]) and TV (TV-Advisor [Das and Horst], PTV [Cotter and Smyth]). TV-Advisor makes use of explicit techniques to generate recommendations for a TV viewer. Such techniques require the user to take the initiative and explicitly specify their interests, in order to get high quality recommendations. Implicit techniques, on the other hand, lessen the burden on the user and try to infer the user's preferences from a viewer's TV viewing history. Explicit questionnaires capture a viewer's TV preferences in a limited manner and only at a coarse level of granularity. Moreover, most of the explicit profile based techniques do not adapt to changing user tastes and are 'static'. PTV uses a content-based plus collaborative filtering approach to generate TV show recommendations. Though they seek similar user profile information like we do, they do not include a 'dynamic', learning algorithm that tracks a person's changing TV preferences over time. We think a dynamic approach to this learning problem is essential to capture the evolving personal TV preferences of a viewer. By incorporating machine learning techniques to connect information pieces embedded in a person's view history, we build a richer, evolving profile for a viewer, over time.

Last year we reported our recommender system work [Gutta, et. al.] based solely on implicit user profiling. In this paper we present the next generation of that system which encapsulates three user information streams--implicit view history, explicit TV viewing preferences, and feedback information on specific shows--into adaptive agents and builds a framework that allows for these multiple agents to collaborate and generate a combined program recommendation for a TV viewer. So far, the setting for this work has been broadcast TV, not yet the PVR.

In the following section we present the architecture of the recommender system and its constituent agents. The section following that will present details of the user test we conducted to evaluate our recommenders' performance. Then we will discuss the results from the user test and finally conclude with a brief discussion on the future directions for our work.

Recommender Architecture

The recommender system architecture is designed for multiple agents to work together and collectively model the profile information on a particular user. The schematic of the system is as shown in Figure 1.

The middle prong in Figure 1 relates to the 'implicit recommender agent'. This is based on the implicit profiles of TV viewers. An implicit profile is built from the viewing history of a TV viewer. The viewing history is a list of TV shows that a viewer has watched (positive examples) and not-watched (negative examples). The not-watched shows are obtained by sampling the database of TV shows that are broadcast in the Briarcliff Manor area (New York City suburb), and are not specified by users. The scheme used for sampling not-watched shows will obviously influence recommender performance. So far, only a scheme that samples one not-watched show uniformly randomly from the week for each watched show has been used. We will study the effect of other schemes in future. The implicit nature of our profiling method stems from the fact that the process does not involve any interaction with TV viewers, regarding their likes and dislikes, other than collecting what shows have been watched. We have developed two implicit recommender agents, one based on Bayesian statistics and one on Decision Trees. For this study they are considered alternatives for implicit recommendations; in future we will explore methods for combining them.

The prong on the right side in Figure 1 is the 'explicit recommender agent'. This agent relies on explicit profiles of TV viewers. These profiles result from a question-answer session with the viewer, wherein the viewers' explicit likes and dislikes towards particular TV channels, show genres and preferred days and times of TV viewing have been gathered. The prong on the left side of Figure 1 is the 'feedback recommender agent'. This agent gathers user feedback on specific TV shows and feeds that information to the implicit and explicit agents.

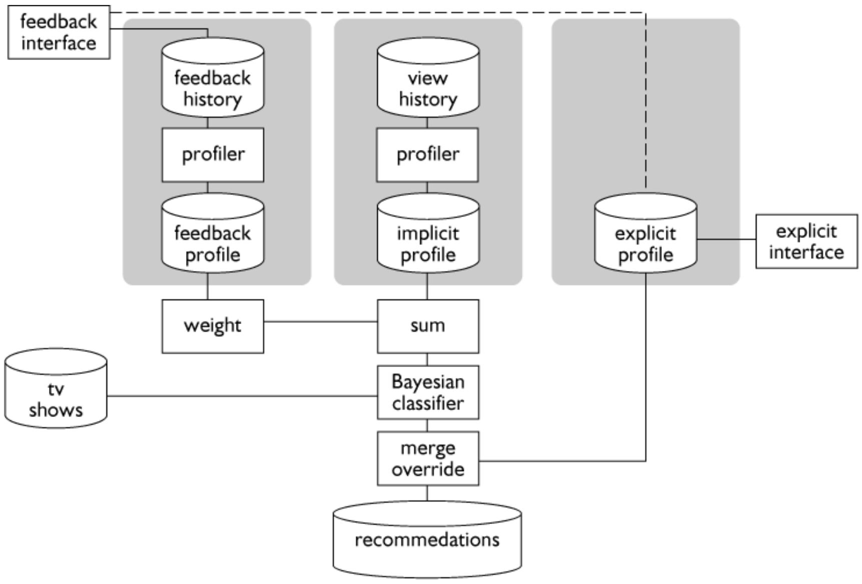

Fig. 1. Recommender system schematic depicting multiple agents and their interactions.

Implicit Recommender Agents

The first of our implicit recommender agents uses the Bayesian classifier [Billsus and Pazzani][Horvitz, et. al.] approach to compute the likelihood that the viewer will like or dislike a particular TV program. We approach the problem with a 2-class Bayesian decision model, where a show either belongs to the class, watched, or the class, not-watched. Ideally, we would like to have ground truth information on whether a TV-show was *liked* or *not liked* by the user, but the implicit profiling technique allows us to have information only on the classes watched and not watched. The user profile, in the Bayesian context, is a collection of features (attribute-value pairs) together with a count of how many times each occurs in positive and negative examples. Let C+ and C- be the classes watched and not-watched. Our Bayesian model first computes the prior probabilities directly from the counts in the viewer profile, as follows:

$T = k(C+) + k(C-)$
$P(C+) = k(C+)/T$
$P(C-) = k(C-)/T$

Then we compute the conditional probabilities that a given feature, f_i, will be present if a show is in class $C+$ or $C-$:

$P(f_i|C+) = k(f_i|C+)/k(C+)$
$P(f_i|C-) = k(f_i|C-)/k(C-)$

The strength of the recommendation is given by $P(C+|\boldsymbol{x}) - P(C-|\boldsymbol{x})$, where $P(C+|\boldsymbol{x})$ and $P(C-|\boldsymbol{x})$ represent the a posteriori probabilities for a new show, given its feature set (vector $\boldsymbol{x}$), that it belongs to a particular class. The a posteriori probabilities are estimated as follows:
$P(C+|\boldsymbol{x}) = P(\boldsymbol{x}|C+)P(C+)/P(\boldsymbol{x})$, where

$$P(x \mid C+) = \prod_{i=1}^{n} P(f_i \mid C+)^{x_i} \quad (1 - P(f_i \mid c+))^{1-x_i}$$

and $P(\boldsymbol{x}) = P(\boldsymbol{x}|C+)P(C+) + P(\boldsymbol{x}|C-)P(C-)$

The second of our implicit recommender agents uses the Decision Tree (DT) approach to compute program recommendation scores. This approach attempts to construct rules for classifying shows given a *training set* of positive and negative shows that are part of the TV-viewing history. We begin by deriving a decision tree (DT) which is then decomposed into rules for classifying the shows. A particular show belongs to one of the two classes: watched or not watched. The decision tree employed uses Quinlan's C4.5 algorithm [Quinlan], which takes an information-theoretical approach based on entropy. This algorithm builds the decision tree using a top-down, divide-and-conquer approach: it first selects an attribute, then divides the training set into subsets characterized by the possible values of the attribute, and follows the same procedure recursively with each subset until no subset contains objects from more than one class. The single-class subsets correspond to the leaves of the decision tree, while a node indicates that a further test needs to be performed on that show to determine which class the show belongs to. When a new show, which is not part of the training set, is encountered, the DT is parsed to obtain a probabilistic class distribution for the show and the class with the highest probability is the predicted class.

Explicit Recommender Agent

The explicit recommender agent takes a person's explicit profile as input and generates program recommendations. The explicit profile comprises a list of features, and their associated user-specified ratings. In the computation of the recommendation score for a show, only the features rated in the explicit profile are relevant.

The explicit score is computed as a weighted sum of the profile

E = W_dt * r_dt + W_ch * r_ch + (1/K) * Σ (W_g_j * r_j, where j = 1..K genres)

where r_dt is the profile rating for the feature daytime (similarly, r_ch for channel and r_g for genre) and the corresponding heuristic weights are: W_dt=0.1, W_ch=0.2 and W_g_j=0.7 (for all genres, j).

Implicit + Explicit Combination Heuristic

In order to compute the combined recommendation score (T) for a show that has an explicit (E) and an implicit (I) recommender score, we define a linear function: T = W_E * E + W_i * I

The weights W_E and W_I sum to one, but vary depending on the difference between E and I. The scheme used weights both equally when E and I agree, but shifts the weighting to favor of E as they become more discrepant (Figure 2).

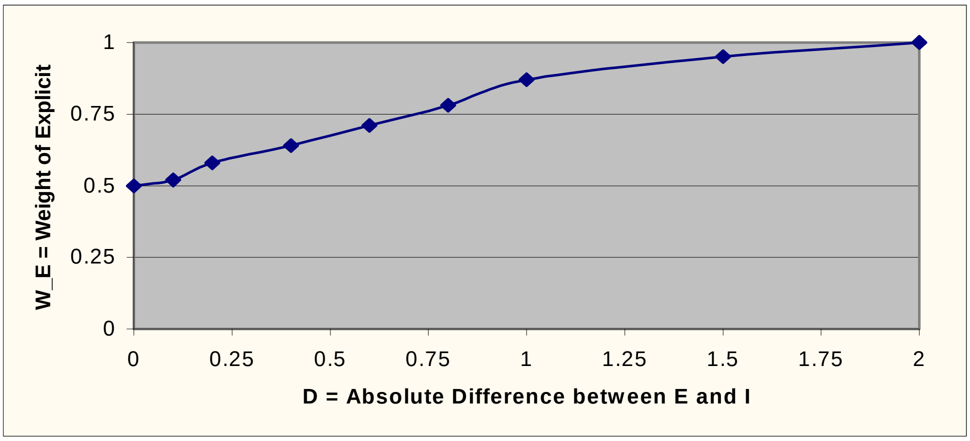

Fig. 2. Empirical relationship between W_E and the absolute difference between E and I scores

Feedback Agent

The feedback agent works in collaboration with the implicit and explicit recommender agents and helps them fine-tune their recommendation quality. The feedback agent presents an opportunity to the user to provide the implicit and explicit recommenders with show-specific information, which the recommenders lacked previously. The design of our adaptation scheme for the feedback agent mirrors the taxonomy of adaptive user interface tasks and agents by Dieterich, et. al. Figure 3 shows the various stages in the feedback agent adaptation process: initiation of the feedback session, proposal of shows to give feedback on, action decisions, and execution of the action. Our feedback agent takes either of 2 forms: one where the system proposes show alternatives (table on the left in Figure 3) and another where the user makes the choice (table on the right in Figure 3) of shows to give feedback on. In the case of system-proposed feedback, the system tries to exploit the fact that 'it knows what it does not know'. The system selects those shows for which a) there is a huge disagreement in scores between the implicit and explicit recommender agents, or b) the implicit agent does not possess information on many of the show's features and hence cannot make an informed prediction about the show, or c) the explicit agent does not posses information on the user ratings for many of the show's features.

	System	User
Initiation of adaptation		●
Proposing show alternatives	●	
Deciding on action		●
Execution of action	●	

	System	User
Initiation of adaptation		●
Proposing show alternatives		●
Deciding on action		●
Execution of action	●	

Fig. 3. Characteristics of the system and user-proposed forms of the feedback agent.

In a feedback session, the user is presented with an opportunity to rate the show (title) itself and key features like actors, actresses, directors, etc., and add them to the explicit profile. When the system seeks explicit information from the user at the start, it cannot know which extra features to seek information on. The feedback agent, by exploiting the knowledge gaps in the system, provides a unique opportunity to seek targeted information and add it to the explicit profile.

The user can give feedback that will have an impact on the implicit recommender agent as well. The implicit profile is augmented with new feature counts based on the user's vote. The feedback agent was not tested in the work reported here.

User Test Design

In order to test the quality of the recommenders we conducted user tests with subjects picked from a variety of living conditions to be representative of a general population. Our test subjects kept paper diaries of the shows they watched. These diaries served as viewing history input to the implicit recommenders with a note for each show flagging those selected by the volunteer or someone else in the household. The number of shows used for training varied from one user to another. We tested 10 users on nine different recommender combinations that were generated from multiple combinations of the implicit and explicit recommender agents and individual and household viewing histories (see Figure 4). The test was designed to answer these questions:

1. Implicit vs. Explicit: is there any difference in the effectiveness of these two approaches?
2. Implicit + Explicit vs. each approach alone: which is more effective?
3. Bayesian vs. Decision Trees: which approach yields better results?
4. Individual vs. Household profiles: are recommendations based on shows the subjects choose themselves (individual) better than those based on all shows they watched but did not choose themselves (household)?

We presented every subject with 9 lists of 30 shows each where the lists corresponded to the 9 recommender combinations. We asked them to give one of the four following responses to the question, "Would you watch this show?": (1) Yes (2) Maybe (3) No (4) I don't know this show. Only the show titles were shown and no other information was given. Each list contained ten highly rated shows, ten lowly rated shows, and ten neutrally rated shows. The recommenders generate a neutral rating when they do not have enough information to discriminate.

We decided against displaying show descriptions and genres with the titles for three reasons. First, users often make viewing decisions from printed program guides based only on a show's title. TV Guide, for example, cannot provide nearly enough space for descriptions of every show. Second, we were worried about test fatigue. If subjects read descriptions for each show, they would lose interest long before they got to the end of the test. Third, we did not want previously viewed episodes to play a role in their decision. A list might present a show they liked and a description of an episode they had seen. Users might choose to mark the show as "No" because they had already seen it.

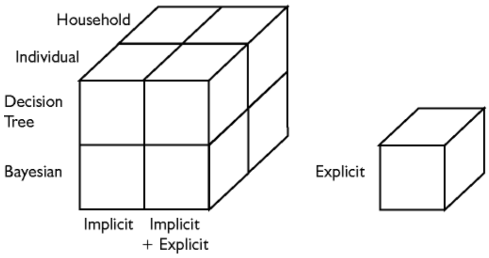

Fig. 4. The nine recommender combinations. Explicit was tested by itself because it only exists for individuals.

Results and Discussion

As our experimental design contained a complete 2x2x2 factorial involving the implicit recommenders and an isolated cell for the explicit recommender, we adopted a two-pronged analysis of variance (ANOVA) approach to the analysis. All significance tests were conducted at the 0.05 level. For the first ANOVA, the factors and levels are shown below in Table 1.

Factors	Levels	Comments
IE	I ie	Implicit recommender alone (i), combined with explicit (ie)
BDT	b dt	Bayesian (b), decision tree (dt)
SH	s h	Self-only view history (s), household view history (h)

Table 1. First ANOVA Analysis

Unfortunately, for three of our 10 subjects we could not employ this full design. One lived alone and so had only a individual view history. In addition, two other subjects had no view histories at all. For these two, we created a composite view history from the household histories of the other subjects. Hence for these three, only a 2x2 design could be applied (IE by BDT). Because of these difficulties, it was decided not to attempt an over-all analysis, but rather to analyze each subject independently.

The outcome analyzed was the square of the difference between the ground truth (GT) value and the recommender score. Since the experiment collected GT ratings for 30 TV shows under each condition, we had 240 data points (n) from each subject (120 for 3 users). Although the 30 shows were selected to have high, medium and low scores for only the recommender condition of one cell, we actually knew the scores for all recommender conditions, so every show actually contributed an error score to every cell in the design. However, we had to first remove all TV shows rated by the subjects as "don't know." This yielded a final “n” that varied from subject to subject.

We emphasize that the missing data (“don’t know” shows) impose an undesirable bias on all the results reported in this paper. The fact that a significant fraction of the shows were unknown to some subjects means that the only ground truth data we could analyze were from known shows. We hypothesize that a significant feature of an implicit recommender is its ability to locate good shows that are unknown to the viewer. If this is true, the analyses below will not show it because unknown shows have been systematically eliminated from the data we analyzed. We plan a follow-up study in which clips from a sample of these unknown shows will be collected and shown to the subjects and ground truth data will be collected for further analyses.

The strongest pattern we observed was that the combination of explicit with implicit was generally better than the implicit recommenders alone. When examining B vs DT main effects, we found B slightly better than DT for 2 users and DT better than B for 1 user and not significant for other users. In each case of a significant main effect, there was an accompanying significant IE BDT interaction and in each case the pattern was the same. The difference between B and DT was significant when the explicit score was not included, but vanished when it was. In other words, combining explicit with implicit recommender scores erased any differences. This is not surprising when we recall that when implicit scores differ from the explicit, the combination heuristic favors the explicit, and does so more strongly as the differences increase.

On analyzing the SH main effect we observed that the errors were smaller while using the household view histories and the implicit recommenders alone, but when combined with explicit, the differences became insignificant. We further observed that DT performed better than B with the household view history, while the view history seems to have no effect on the Bayesian. Again, all differences vanish when the explicit scores are combined, but the Bayesian alone may be better still.

For the second ANOVA, a one-way design was selected where each recommender condition was treated as a separate cell. The nine levels are shown below in Table 2. The post-hoc contrasts were computed using the Tukey B method [Cleary and Yang].

Levels	Comments
i b h	Implicit Bayesian using household view history
i b s	Implicit Bayesian using individual view history
i dt h	Implicit DT using household view history
i dt s	Implicit DT using individual view history
E	Explicit recommender
Ie b h	Combined explicit implicit Bayesian using household view history
Ie b s	Combined explicit implicit Bayesian using individual view history
Ie dt h	Combined explicit implicit DT using household view history
Ie dt s	Combined explicit implicit DT using individual view history

Table 2. Second ANOVA Analysis

The analysis of the one-way design reinforced our earlier conclusions for the two-way design. Specifically we observed that the explicit was better than the implicit recommenders for most of the users. When not combined with the explicit scores, which washes out the differences, the Bayesian seems to have a slight advantage over decision tree.

Conclusions and Future Work

The combination of implicit and explicit agents seemed to work the best in our multi-agent system. We would like to point out that the effect of the feedback agent on the other two has not been tested thoroughly at this juncture. The explicit agent alone did well in comparison to both the implicit recommender agents. We believe this is partly due to the nature of the test: the explicit recommender is based on what the subject tells us they like and the test results are also based on what shows subjects tell us they like. So it makes sense that the subjects' responses agree more with the explicit recommender. We plan to undertake another user test in the near future to reveal the subjects' preferences for shows that they have not seen earlier. Contrary to our expectations, the individual user profile did not make any difference in the recommender accuracy over the household profile. This test needs to be repeated over a much larger sample in order to draw concrete conclusions regarding the necessity to maintain individual or household profiles in a PVR.

We have the following activities planned as part of our future work:

1. Experiment with other combination heuristics for combining the scores from multiple agents.
2. Investigate the following aspects of the recommender: a) effect of the length of viewing history on the recommender accuracy, b) choice of weights in the explicit recommender agent, and c) the effect of feedback on the learning rates of other agents.

References

Billsus, D. and Pazzani, M. Revising User Profiles: The Search for Interesting Web Sites. In *Proceedings of the Third International Workshop on Multistrategy Learning*. AAAI Press, 1996.

Cleary, P., A., and Yang, Y., K. Comments on Statistical Procedures. *ECDEU Assessment Handbook.* US Department of Health Education and Welfare, 1976.

Cotter, P., and Smyth, B., PTV: Personalised TV Guides. In *Proceedings of the 12th Conference on Innovative Applications of Artificial Intelligence*, IAAI 2000, Austin, Texas, 2000.

Das, D. and ter Horst, H. Recommender Systems for TV. In *Proceedings of AAAI*, 1998.

Dieterich, H., Malinowski, U., Kuhme, T., and Schneider-Hufschmidt, M. State of the Art in Adaptive User Interfaces. Schneider-Hufschmidt, M., Kuhme, T., Malinowski, U. (Eds.), in *Adaptive User Interfaces, Principles and Practice,* Elsevier Science Publishers, Amsterdam, The Netherlands, 1993.

Gutta, S., Kurapati, K., Lee, K., P., Martino, J., Schaffer, D., and Zimmerman, J. TV Content Recommender System. In *Proceedings of Seventeenth National Conference on Artificial Intelligence*, pp. 1121-1122, AAAI, Austin, July 2000.

Horvitz, E., Breeze, J., Heckerman, D., Hovel, D., and Rommelse, K. The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, July 1998.

Lieberman, H. Autonomous Interface Agents. In *Proceedings of CHI*. Atlanta, March, 1997.

Lashkari, Y., Metral, M., and Maes, P. Collaborative Interface Agents. In *Proceedings of the National Conference on Artificial Intelligence*. AAAI, Seattle, August 1994.

Pazzani, J., Muramatsu, M., and Billsus, D. Syskill and Webert: Identifying Interesting Web Sites. In *Proceedings of the Thirteenth National Conference of AAAI*, pp. 54-61. 1996.

Quinlan, J. R. Learning Efficient Classification Procedures and their Application to Chess End Games. Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. (Eds.), in *Machine Learning: An Artificial Approach*. Vol. 1, Morgan Kaufmann Publishers Inc., Palo Alto, California, 1983.

Sheth, B. and Maes, P. Evolving Agents for Personalized Information Filtering. In *Proceedings of Ninth Conference on Artificial Intelligence for Applications*, IEEE Computer Press, March 1993.