On Learning Video Browsing Behavior from User Interactions

Brian Westphal
Dept. of Computer Science
University of Reno, Nevada

Tanveer Syeda-Mahmood
IBM Almaden Research Center
650 Harry Road, San Jose 95120
stf@almaden.ibm.com

Abstract

With e-business becoming mainstream, inferring intentions of users by observing their interactions over the web, becomes an important problem. Current methods of customer tracking using simple and heuristic data categorization methods, are insufficient to model the complex time-varying behavior of users. In this paper, we continuously infer and track the browsing behavioral states of users through the use of Hidden Markov Models (HMMs). Specifically, we model the behavioral states a user transitions while browsing, to be the hidden states of a Hidden Markov Model. We develop a systematic way to design and estimate the parameters of the behavioral HMMs using a combination of supervised and unsupervised learning with data obtained from user studies. The design of behavioral HMMs is illustrated in the context of video browsing.

1 Introduction

With e-business becoming mainstream, inferring intentions of users by observing their interactions over the web, becomes an important problem. As the cost of acquiring online customers increases, e-businesses are looking for attentive systems that work cooperatively with users, learning their interests and facilitating their needs and goals. To enable this, however, such attentive systems must be equipped with tools to do dynamic user behavioral modeling.

Learning browsing behavior from interaction patterns, however, is a difficult problem. First, it is not clear, if a single interaction can reveal a browsing state. Next, past browsing states clearly have an effect on current browsing states, and must be taken into account. Finally, the same sequence of observations could imply multiple browsing states. Clearly, simple if-then-else rules cannot effectively deal with such uncertainties associated with user interactions. More complex machine learning-based user models such as Bayesian networks[3, 8], on the other hand, may require a complex design by hand, with an a priori anticipation of all possible use cases to ensure good initialization. What is desirable is a simple, yet powerful learning framework that can accurately model the complexity of such time varying behavior. Further, it should be possible to design it with ease, and should have good methods to initialize and learn based on simple user feedback.

In this paper, we use a theoretically well-founded methodology based on Hidden Markov Models (HMMs) for learning browsing behavior. Using the HMM, a sequence of user interactions with Web content are treated as an observation sequence generated by the model. The behavioral states are then the hidden states of the HMM. Learning the behavior of each user involves designing an HMM that can explain his/her sequence of interactions in an 'optimal' sense. Tracking the behavior of users then reduces to predicting the instantaneous browsing states of users using the HMM and the given observation sequence.

We illustrate the design of behavioral HMMs in the context of video browsing. With thousands of hours of content becoming available on demand, video browsing has become common for news, instruction, advertisement, and entertainment. By observing the interaction patterns of users during video browsing, media players and streaming media servers can learn what interests users. This is potentially valuable not only for improved customer tracking and context-sensitive e-commerce, but can also be useful in the generation of fast previews of videos for easy pre-downloads. To illustrate the design of HMMs for video browsing, we designed new media players to record and learn from user interactions. By noting the underlying video content in browsing states indicating interest, interesting video content was automatically identified. The learning and interest tracking module has been thoroughly tested and integrated into a streaming media server (MediaMiner) and a media player client (VideoChargerPlus) product prototype.

While HMMs have been used before for modeling and predicting different kinds of web browsing behavior[7, 9, 1], and for plan recognition[8, 4, 3, 5, 2], the use of HMM to predict the interestingness of video from deciphering the interest level of users is novel. Also, our approach uses supervised learning to automatically initialize HMM parameters unlike other manual approaches.

Any application can thus be equipped with user modeling through a two-step process. In the training stage, users specify their behavioral states at randomly chosen instants. The resulting state-label pairs is used to train a basic HMM. The parameters of the HMM are then refined for each individual user during operation by observing their interaction sequence and updating using the EM algorithm.

#	Genre	#	predictions		
		observed	correct	false	% agreement
1.	Lecture	2536	2104	432	82.9%
2.	documentary	1732	1560	170	90.06%
3.	documentary	1056	902	154	85.4%
4.	Instructional	2475	2380	95	96.9%
5.	Instructional	3347	3095	252	92.4%
6.	Talk/Seminar	3084	2839	245	92.5%

Table 1: Illustration of effectiveness of browsing state prediction accuracy of MediaMiner for different video types.

2 Modeling browsing behavior using HMM

It is reasonable to assume that the action elicited by a user is a reflection of his current behavioral state. Further, we can also make the assumption that the current behavioral state at time 't' is a function of the past behavioral states. This implies that the influence of the past behavioral states on the current behavioral state is a Markov chain. Thus browsing states can be modeled through the mechanism of a Hidden Markov Model[6]. A Hidden Markov Model is a stochastic signal model to describe a system that can be in one of a set of N possible states $\{S_1, S_2, ..S_N\}$ at any time instant 't', and produces one of a set of M possible observation symbols $\{V_1, V_2, ...V_M\}$ at 't' see Figure 3a). The transition between states, as well as the probability of seeing a specific observation symbol while in a state are described through probability distributions. Specifically, if we restrict attention to a first order HMM, it is characterized by

$$\lambda = (M, N, A, B, \pi) \tag{1}$$

where $A = \{a_{ij} | 1 \leq i, j, \leq N\}$ is the state transition probability distribution, $B = \{b_j(k) | 1 \leq j \leq N, 1 \leq k \leq M\}$ is the observation symbol distribution, $\pi = \{\pi_i | 1 \leq i \leq N\}$ is the initial state distribution [6] (See Figure 3a), and

$$\begin{aligned} a_{ij} &= P(q_t = S_j | q_{t-1} = S_i) & (2) \\ b_j(k) &= P(O_t = V_k | q_t = S_j) & (3) \\ \pi_i &= P(q_1 = S_i) & (4) \end{aligned}$$

where O_t is the observation symbol seen in the observation sequence at time t. Here we assume a first-order Markov chain, so that that $P(q_t | q_T, q_{T-2}, ...q_{t+1}, q_{t-1}, ...q_1) = P(q_t | q_{t-1})$ for a T-length observation sequence $O = \{O_1, O_2, ..O_T\}$.

2.1 Designing behavioral HMMs

From Equation 1, it is clear that to design the behavioral HMMs, we need to know (a) the observation symbols, (b) the browsing states and their labels, and (c) the estimates of the model parameters (A, B, π). The observation symbols in most cases are a function of the user interface with obvious choices being names of buttons pressed, or mouse actions. Determining the number of states of a HMM is still an open research problem. In the context of user modeling, there is the additional problem of what to call these states, a process best done manually. The number of states is expected to be different for different user behavioral tasks. For modeling browsing behavior, we propose that a set of initial browsing states be manually identified based on prior experience with browsing interactions over the web. The final list of states, can then be derived by conducting user studies in which users are asked to either concur with one of the existing choices or enter their choice of browsing states during their interactions. These studies in the context of video browsing are described in detail in Section 3.1. As an example, the set of observation symbols and browsing states assembled through such studies are summarized in Table 2. We believe these state labels are also applicable to browsing of other content on the Web.

Once the observation symbols and browsing states are determined, the HMM parameters can be learned using an unsupervised learning algorithm, such as the EM algorithm as described in standard text books on HMM and papers[6].

2.2 Predicting the browsing states

Once the HMM model has been estimated, learning the browsing state of a user at any time instant t is equivalent to determining the most likely hidden state at time $q_t \in Q = \{Q_1, Q_2, ...Q_T\}$ given the observation sequence O and the model λ, that is optimal in some meaningful sense, i.e., best "explains" the observations. Again, this is a well-known problem in HMM with several optimality criteria available. The popular algorithm is called Viterbi algorithm and is described in several books and papers including [6].

2.3 Training the HMM

The EM or the Baum-Welch algorithm can iteratively estimate the model parameters starting from initial estimates and is an example of unsupervised learning. Such maximization, however, leads to local maxima only, with the nature of the maximum reached critically dependent on the choice of the initial estimates. To enable effective learning, we augmented this unsupervised learning by a training or supervised learning stage in which manually chosen pairs of observation sequence and state labels were used to generate the initial estimate in two steps. In the first step, we initialized the HMM parameters by making some obvious impossible state and symbol transitions have zero probabilities. The rest of the parameters were obtained from uniform distributions. In the second step, we used results from user studies in which training date was obtained by allowing users to specify their state at random time instants during their browsing sessions. The experimental setup and the focused tasks that elicited the various browsing behaviors are described in detail in Section 3.1. The observation symbols of each session were recorded continuously giving rise to observation sequences with browsing states at random instants.

From this data, the initial estimates of the parameters were then derived as:

$$\pi_i = \frac{\text{Number of times } q_1 = S_i}{\text{Total number of observation sequences}} \tag{5}$$

To compute a_{ij}, we noted pairs of consecutive observation symbols for which the state label were known to give

$$a_{ij} = \frac{\text{Number of times the pair } q_t - 1 = S_i \text{ and } q_t = S_j}{\text{Total number of labeled consecutive state pairs}} \tag{6}$$

Finally, $b_j(k)$ is obtained by noting for each occurrence of a state label S_j, the fraction of times the observation symbol V_k was noted.

3 Learning framework applied to video browsing

We now illustrate the learning framework in the context of video browsing. The goal here is to observe the sequence of interactions made by viewers of video and infer their browsing states. Ordinary media players do not offer sufficient controls to record or elicit a wide variety of browsing behavior. For this reason, we designed new media players (based on RealPlayer 8 and IBM's VideoCharger) that offer in addition to the usual play, pause and random seek, an ability to visually browse through the video faster as well as slower, forward as well as backward with variable speed. With the controls offered, a richer set of browsing behaviors could be elicited from viewers and the resulting set of observation symbols is shown in Table 2. For different media players, however, this list can be different. The video browsing states and HMM initialization was done using information derived from user studies described below. Continuous learning and prediction of HMM parameters was done as described in the previous section.

3.1 User studies

We conducted a series of user studies with the twin goals of a) collecting data for training the behavioral HMM, and (b) collecting data to test the effectiveness and utility of behavioral modeling. Specifically, Study1 required viewers to watch video and label their browsing states at randomly chosen time instants. This information is used to train the HMM. Study2 also required viewers to watch video, but this time, they indicate agreement or disagreement with the learning module's prediction of browsing state. This data helps in evaluating the accuracy of prediction.

The subjects for the different user studies all came from our lab, with 10 people taking part in Study1, 20 in Study2, and 10 new users in Study3. The design of the studies was done in conjunction with a UI designer and three video experts. The users recruited were note previously familiar with the video content. A set of 20 videos was assembled, of which 6 were set apart for training. The videos depicted a variety of genres varying from instructional, talk videos to entertainment videos and documentaries. The length of the videos varied from as short as 9 minutes to as long as 1 hour and 30 minutes.

Each of the videos was watched by three video experts who were then asked to come to a consensus on the topic of the video, and write a summary of up to two paragraphs for the video. They were also asked to identify three to four questions whose answers could only be found by watching the video. The selection of questions was from "interesting" video regions and were designed to elicit behavioral patterns that may be associated with searching for content. Finally, the questions ensured that the recorded observation sequence would show a good mix of the observation symbols from Table 2.

Study1:

To obtain the data for training the HMM, a set of initial browsing states were hand-selected. These were the states S1 to S5, and S9 of Table 2. Users were asked at randomly chosen time instants to enter their current state while making an interaction. Each new state thus assembled was also added to the list of choices for subsequent selections. At the end of Study1, the list of user identified browsing state labels were consolidated to obtain the list shown in Table 2. Some new states derived during this process include S9 ('did-not-find-what-I-wanted' state).

Study2:

For user study2, the system continuously recorded the observation sequences, and predicted the browsing states of users based on the initially trained HMM and its successive refinement for each new observation sequence using the EM algorithm as described in Sections 2. The correctness of prediction was recorded through random sampling of users as described above. The data obtained from the corrected states was not used for re-initialization of the HMM to avoid testing bias.

4 Results

To evaluate the browsing state prediction made by the learning system, we used the data collected from Study2. That is, of the 13,000 of the 25,000 observations where ground truth data was obtained in Study2, we recorded the number of times users agreed with the learning system's prediction of browsing states. We also recorded the continuous prediction of browsing behavior within a single viewing session of a user. The resulting instantaneous predictions are illustrated in Figure 3c. These browsing state predictions of the system are shown connected by dark lines. The state labels marked by the viewer at randomly chosen time instants are shown overlaid on the predicted state sequence through cross-marks in Figure 3c. Notice although there are a few mismatches, the state prediction agrees, in most part, with the browsing state chosen by the viewer. Collecting over 25,000 observations from all user sessions with all videos tested, and retaining the state predictions for which we had ground truth labels, the percentage agreement of the prediction for different videos is indicated in Table 1, where columns 4 and 5 indicate the number of correct and false predictions for a sampling of the videos of different genres tested. As can be seen, the disagreement is relatively small and is proportionately similar for all video genres tested.

Mining browsing data at servers

The distribution of browsing states across users per video was recorded. These distributions confirmed that most users of video were in 'looking-for-something' state as shown in Figure 4a and b for two of the videos tested. This figure also indicates the most users did find what they were looking for (state S7). Similarly, if the predicted state for most viewers for a specific video was S4 or "abandoned interest" then it can be concluded that the video was not watched by most users, a piece of information that may be of interest to several media tracking companies. Similarly, in a way analogous to click stream analysis, distributions of observation symbols generated during user interactions with a single video were recorded. The distribution of these symbols for two of the videos are illustrated in Figure 4a and b. Distributions such as these can help evaluate user interfaces. For example, from these figures, we could conclude that most users preferred the fast forward feature of the player over slow forward.

Determining interesting content

The browsing state information is not only useful to infer the behavior of users, but can also help in identifying interesting content. Thus if it can be determined that the viewers were in an interested state (S3 or S7 in our case) while watching a specific section of the video, then the underlying video segment can be inferred to be interesting. The distribution of interesting segments can give additional information over simple video usage distribution that can be obtained from conventional server log analysis. In addition, by assembling the interesting segments together, a short summary or preview of the video can be obtained. Such user-interest-based previews can then be made available to viewers as pre-downloads thus easing network traffic.

We first illustrate the automatic detection of interesting segments based on the knowledge of behavioral states. Figure 1b shows the distribution of visit counts of all video time instants that were visited in states S3 or S7 for one of the videos tested during Study2. As expected, we see distinct peaks in this figure. We also note that a number of video segments are not touched at all in these browsing states. In contrast, a visit count of all the video time instants visited without regard to the browsing states shows a distinctly different distribution spanning the entire video as shown in Figure 1a (for the same video). By selecting regions around the peaks and ranking them, we get the interesting segments detected by the algorithm for this video as those shown in Figure 1c. Figures 1e-h shows a similar detection of interesting segments for a second video. Here again, difference between visit count with and without knowledge of browsing states can be seen in Figure 1e and f.

Since all videos had their interesting segments labeled by the experts, the difference in the automatically detected and manually identified interesting segments could be recorded. Figure 1 illustrates the resulting overlap between the predicted and actual interesting segments for two of the videos tested. As can be seen from these figures, the number of automatically detected interesting segments is usually higher than the actual number of such segments. But the ground truth durations were always included within one of the interesting segments detected by the algorithm (shown in blue respectively in the two figures). The automatically determined segments are also larger indicating the characteristic trial and error search used by viewers for zooming onto the correct sections of the video.

This was found to be true on repeating the experiments over the entire suite of 20 videos. The extra interesting segments detected are usually due to one of two causes: (a) algorithm errors in state prediction (b) viewer's engaging in browsing behavior exhibiting either curiosity or misjudgment of the durations in the video containing the answers.

5 Summary

In this paper, we have presented a learning framework for inferring browsing behavior based on HMMs. Through extensive user studies and experiments, we have shown that such learning models not only model user behavior accurately, but are also

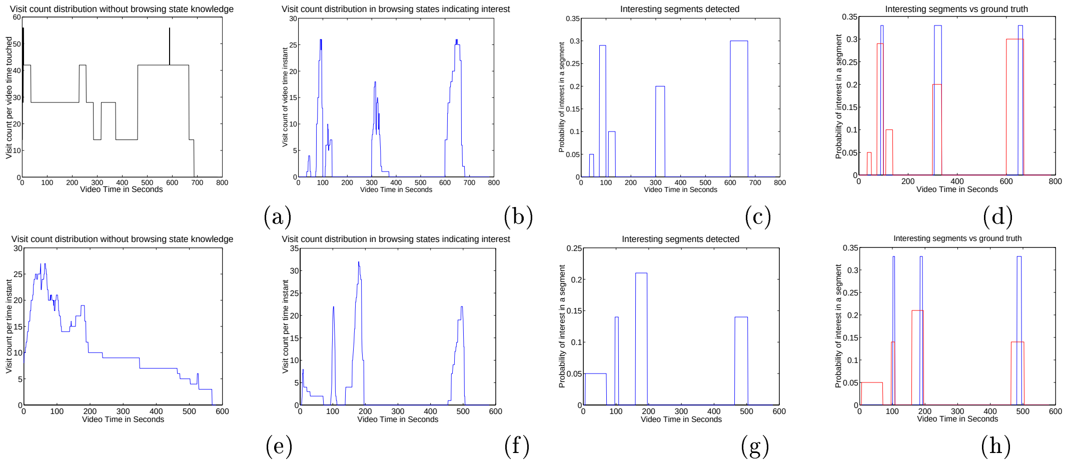

Figure 1: Illustration of detection of interesting video segments. (a) Distribution of segments viewed collected across all users for a single video without using the knowledge of browsing state. (b) Distribution of selected segments viewed in browsing states indicating interest collected across all users for the same video. (c) Distribution of ranked interesting segments detected by the algorithm. (d) Comparing manually detected interesting segments with automatically extracted interesting segments. Here the manually detected segments are shown in blue.(e)-(h) Same as (a) - (d) for a second video.

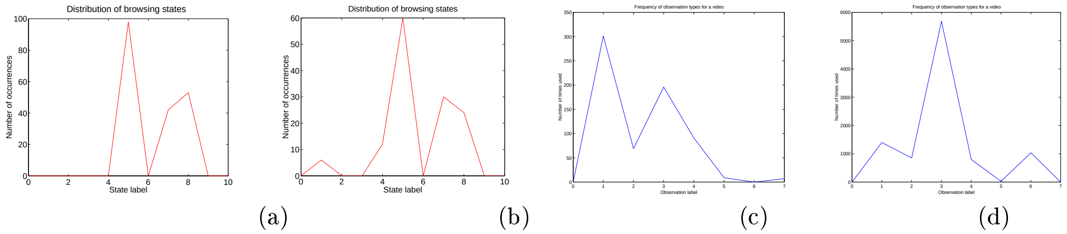

Figure 2: Illustration of state label and observation label distribution. Distribution of browsing states predicted by MediaMiner across all viewing sessions and observations for a single video. (b) Same as (a) for video 2. (c) Distribution of observation symbols recorded by MediaMiner across all viewing sessions and observations for a single video. (d) Same and (c) for video 2.

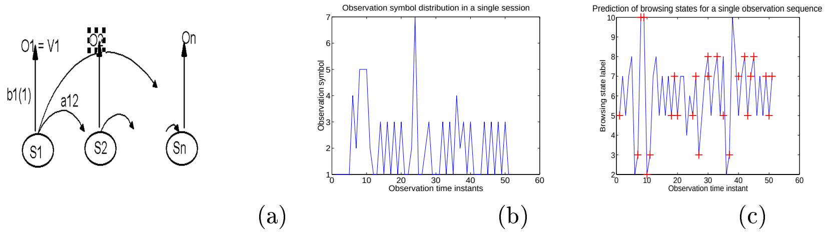

Figure 3: Illustration of continuous prediction of browsing states using HMM. (a) A simple description of an HMM (b) A single user's observation sequence generated during a viewing session of a single video in the training mode. (c) The predicted browsing states at corresponding observation times.

State label	Description	Observation symbol	Description
S1	Curious	O1	Play
S2	Aimless browse	O2	Pause
S3	Found something interesting	O3	Slider
S4	Abandoned interest	O4	Fast Forward
S5	Looking for something	O5	Fast Reverse
S6	Resumed interest	O6	Slow Forward
S7	Found what I wanted	O7	Slow Reverse
S8	Undetermined		
S9	None of the above		
S10	Did not find what I wanted		

Table 2: Illustration of browsing state labels and observations derived from the user study during MediaMiner training. The observation symbols are specific to the media player used in MediaMiner.

useful from annotating the content's perspective. Although the design of the HMM has been illustrated for the application of video browsing, similar methodology can be used to design HMMs for Web browsing.

References

[1] I. Cadez et al. Visualization of navigation patterns on a web site using model-based clustering. Technical report, Microsoft Research, March 2000.

[2] B. Davison and H. Hirsh. Experiments in unix command prediction. Technical report, Technical Report ML-TR-41, Dept. of Computer Science, Rutgers University, August 1997.

[3] E. Horvitz et al. 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*, pages 256–265, 1998.

[4] J-J Lee and R. McCartney. Plan recognition in human computer interaction. In *Proceedings of the Plan Recognition Workshop, IJCAI-95*, 1995.

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

[6] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings IEEE*, 77(2):257–286, 1989.

[7] R. Sarukkai. Link prediction and path analysis using markov chains. In *11th World-wide Web Conference*, 2000.

[8] A. Tomoyosi and H. Tanaka. A bayesian approach for user modeling in dialogue systems. Technical report, Dept. of Computer Science, Tokyo Inst. Technology, ISSN 0918-2802, August 1994.

[9] D. Zhang and Y. Dong. An efficient algorithm to rank web resources. In *11th World-wide Web Conference*, 2000.