Classification and tracking of hypermedia navigation patterns

Patrick Gallinari, Sylvain Bidel, Laurent Lemoine, Frédéric Piat, Thierry Artières

LIP6, Université Paris 6, 8 rue du capitaine Scott, 75015, Paris, France
fname.name@lip6.fr

Abstract. We consider the classification and tracking of user navigation patterns for closed world hypermedia. We first propose a series of features characterizing different aspects of the navigation behavior. We then develop Hidden Markov models and a variant of these models called Multi-stream Hidden Markov models to track on line the behavior of a user. We also provide experimental results for the recognition of pre-defined user behaviors, using a home made basis.

1 Introduction

The development and the complexity increase of hypermedia systems accessible by many different users (e.g. through internet) has created a need for developing help tools for the user. Basic mechanisms relying on navigation history or dedicated search engines are already of common use but are insufficient. To go further it is necessary to develop user centric help strategies then to characterize the user in order to define help actions, to infer the relevant action for a user in a given situation [2]. The user modeling field has explored such issues for many years now, some aspects of the problem are already pretty well covered by existing solutions, while many other are still largely open. One reason for that is that the domain is rapidly evolving and adapts continuously to the technology. Another one is that most aspects of human behavior, like measuring satisfaction, inferring user goal, etc, are intrinsically complex and/or subjective, and this difficulty carries to the development of user modeling tools.

We focus here on the characterization of user groups or categories with respect to user navigation in a rich hypermedia system, from the observation of low level traces like clics, scrolls, page access, etc. The goal is to follow individual user navigation and to track its behavior during a session so that an adequate help could be provided to him on line. There have been some studies for defining generic navigation behaviors in hypermedia e.g. [4]. Most agree on 5 or 6 typical behaviors and even if they propose different classifications, they globally offer a coherent vision of typical user behaviors. In our work we use a behavior taxonomy in the different steps of the system development: data gathering, action sequences classification, behavior tracking, behavior interpretation.

We first propose an original set of features for characterizing the different aspects of user navigation from traces in rich hypermedia systems. The sequence of user actions captured by a server is then encoded in a sequence of feature vectors or frames, each frame representing navigation actions for a short duration. For identifying navigation behaviors, we then make use of Hidden Markov Models (HMMs). We also introduce the Multi-Stream HMM (MS-HMM) Model that allows taking into account simultaneously different partially asynchronous feature sets characterizing the user behavior at different time scales. We use these models for user behavior categorization and tracking. Up to

now, sequence models – either Markovian models [10] or dynamic Bayesian Networks [8] have mainly been used for predicting user actions or for inferring goals in environments which are described using existing domain specific knowledge.
We first discuss in section 2 the navigation behaviors we want to recognize, the database used in our experiments and the features we propose for characterizing user sequences. In section 3, we introduce sequence models that operate on these feature and present experimental results in section 4.

2 Typical navigation and Feature Extraction

2.1 Navigation typology

Although most researchers who proposed typologies of navigation behaviors and search strategies in closed hypermedia systems distinguish broad user strategies (e.g. browsing and searching), there is no general agreement for the generic categorization of more specific behaviors. We adopt here the taxonomy proposed by Canter and al [4] which we found convenient for the type of application we deal with, and adapt it to our problem, we thus distinguish four *elementary behaviors*:

- Scanning: seeking an overview of a theme by requesting an important proportion of its pages but without spending much time on them.
- Exploring: reading thoroughly the pages viewed.
- Searching: seeking a particular document or information.
- Wandering: navigating in an unstructured way without particular goal.

2.2 Navigation Database

For the tests, we used the application « The XXth century encyclopedia », initially a cultural CD-ROM[1] reconfigured as an Internet site. This is a typical "cultural" hypermedia system, it contains about 2k articles (i.e. pages with text, pictures, videos etc…), a full-text search engine and tables of contents where the user can navigate on a 2-level theme hierarchy. Each theme is associated a set of key words. Each article is associated a theme, navigation links towards other articles, and reading times corresponding to the durations required to fully read each of its paragraphs. All the above have been set by the conceivers of the site.
In order to evaluate our methods, we have generated data in a controlled fashion, by asking 26 users to fill out questionnaires by navigating through the encyclopedia. The questions were chosen in order to induce a given navigation behavior from the above typology. For instance, a question asks the user to extract important dates from a particular theme. This leads the user to view several pages of this theme without having to read them thoroughly, it is a "Scanning" behavior. For the "Exploring" behavior, the user was asked to fully read a few articles (pages) etc. The sequences of user actions recorded by the navigator are labelled by the typical behavior expectedly induced. These labels will be used for the evaluation. We gathered 104 data sessions, 26 for each of the

[1] Distributed by Montparnasse Multimédia company.

4 *elementary behaviors*. Navigation data are sequences of dated events (page access, click, scroll, query on the search engine...) that are collected all along the user session.

2.3 Navigation features

Traces are then processed to compute sequences of feature vectors. We computed a frame for every page viewed. Overall, this yields over 400 frames, each representing a period of about one minute in average. We investigated various features and in the end, we are left with 9 features that take advantage of the richness of the information associated to articles (reading time, etc..). These 9 features are divided into three subsets according to the type of information they carry:

- The "reading" subset reflects the extent and the quality of the reading behavior and contains 4 features. We use here the reference reading time for each paragraph, to compute reading rates for the first quarter of the document and for the rest of the document (applies when accessed via scrolling). The time spent on the page(s) and the activity (number of clicks/scroll events) complete this set of features.
- The "resources" subset informs the system about the kind of resources used. They may be articles (the real content of the hypermedia, the leave pages in the tree of themes hierarchy), tables of content (either of 3 levels, containing links to access the themes, sub-themes or articles), or the search engine page. We use the percentage of time spent on these three kinds of resources.
- The "navigation" subset characterizes the navigation focus, whether the user is focused on one theme or spread onto several. We define the distance between two articles as being the distance between their sub-themes in the tree. We then compute the length of the path followed, and the standard deviation of the distances between the pages visited and the focus sub-theme (with the most time spent on).

3 Behavior Models

After the feature extraction step, the navigation information in a user session is represented as a sequence of frames (a frame is a vector of 9 features), each frame corresponds to timely information about the user actions. Let $o_1^T=(o_1, \ldots, o_T)$ denote a sequence of T frames, o_t being the t^{th} frame in the sequence. We want to identify different types of user behavior, for that we propose a model for the production of frame sequences. This model B will then allow to compute sequence likelihood $P(o_1^T/B)$. We investigated for that two Markovian systems and, as a reference system for the supervised case only, a Multi-Layer Perceptron (MLP).

The MLP is trained (using Back Propagation algorithm) to discriminate between the frames corresponding to different behaviors. It takes a frame as input and outputs a vector of behavior scores, the maximal score corresponds to the recognized behavior. When trained for discrimination, a MLP is known to approximate posterior probabilities $P(B/o_t)$. Then one can use this MLP to classify sequences of frames since, using Bayes Theorem and assuming uniform behaviors priors:

$$\arg\max_B P(o_1^T / B) = \arg\max_B \prod_t P(B / o_t) \tag{1}$$

Besides, we developed two Markovian systems. The first is based on standard HMMs which have shown strong abilities for various signal modelling and classification tasks.

We used one HMM per behavior, with an ergodic topology (any transition allowed) and diagonal covariance Gaussian densities. The underlying hypothesis for HMMs is that the process being modeled is locally stationary and a transition in the Markov model corresponds to a skip from one of its stationary state to another one. A consequence is that all features in the frames are assumed locally stationary and synchronous processes. This assumption does not correspond to the features used here which do not change synchronously. Hence, we propose to use a variant of HMMs called Multi-Stream HMM (MS-HMM) [9], it allows combining multiple partially synchronous information streams or modalities [6, 7]. More precisely in our case, a behavior model is a combination of three HMMs operating on different information stream corresponding to *Reading*, *Ressources* and *Navigation* frame sequences (see §2). The three streams are asynchronous i.e. transitions in the three stream-HMMs may occur at different times, except in some particular states named recombination states that are designed by hand. We choose the entering and leaving states of each stream model as recombination states, i.e. each behavior model is fully asynchronous. This means that, given an entering time and a leaving time in a stream behavior model, one can compute very simply the probability of the corresponding sub-sequence of frames. For example, the probability of a sub-sequence of frames from time b to time e is computed with:

$$P\left(o_b^e / B\right) = P\left(rd_b^e / B_{rd}\right) P\left(rs_b^e / B_{rd}\right) P\left(n_b^e / B_n\right) \qquad \textbf{(2)}$$

where B is a behavior MS-HMM model composed of three HMM model B_{rd}, B_{rs}, B_n, working respectively on sequences of frames rd_b^e, rs_b^e, n_b^e which are frames of *reading*, *resources* and *navigation* features. Recombination states will be useful for segmentation tasks as will be seen in §3.3.

3.1 Supervised and unsupervised learning

For our experiments, we investigated supervised and unsupervised learning. Supervised experiments aim at investigating the ability to correctly classify and track *elementary behaviours*. Unsupervised experiments aim at investigating the ability to automatically discover typical user behavior from a collection of unlabelled data (user traces), then to classify and track these behaviors.

For supervised learning, we used the labelling of our database into the four elementary behaviors as described in §2. We learned four models, one for each behavior, using a classical learning scheme where each behavior model is trained to maximize the likelihood of associated training sessions.

For unsupervised learning, we consider a mixture of N probabilistic (Markovian only) behavior models. The probability of a frames sequence is given by a mixture of models:

$$P\left(o_1^T\right) = \sum_{i=1..N} P(B_i).P\left(o_1^T / B_i\right) \qquad \textbf{(3)}$$

where $(B_i)_{i=1..N}$ are N Behavior models, $P(B_i)$ is the prior probability for the ith behavior model B_i and $P\left(o_1^T / B_i\right)$ is the likelihood of o_1^T computed by B_i (HMM or MS-HMM). Learning consists in maximizing the likelihood of all training sessions given this mixture model. Since we do not know which behavior a training session belongs to, we use an EM procedure where missing data are posterior probabilities of behavior $P\left(B_i / o_1^T\right)$. Here is the sketch of the algorithm, it is close to the one in [3]:

0. Initialise the parameters of all behavior models $(B_i)_{i=1..N}$ and of priors.
1. Iterate until convergence
   i. Estimate missing data using current models.

$$P\left(B_i / o_1^T\right) = \frac{P\left(o_1^T / B_i\right)P(B_i)}{\sum_{j=1}^{N} P\left(o_1^T / B_j\right)P\left(B_j\right)} \quad \textbf{(4)}$$

   ii. Re-estimate behavior models with all training sessions. A session o_1^T participates to the re-estimation of model B_i with a weight corresponding to $P\left(B_i / o_1^T\right)$.
   iii. Re-estimate behavior models priors:

$$P(B_i) = \frac{1}{\#Training\ sessions} \sum_{o_1^T \in TrainigData} P\left(B_i / o_1^T\right) \quad \textbf{(5)}$$

3.2 Behavior categorization

When sessions correspond to a single *elementary behavior* as it is the case for the recorded sessions (§2), it is useful to categorize whole sessions into one of the 4 *elementary behaviors*. This amounts to classify sequences using HMMs or MS-HMMs: a session is classified according to the model maximizing the sequence likelihood.

3.3 Behavior segmentation

When sessions correspond to multiple successive behaviors, we will try to detect the sequence of navigation behaviors of a user along a session. A global Markov model is then built by concatenating the leaving state of each behavior model to the entering state of each behavior model. Then, considering a test session, a dynamic programming algorithm finds the optimal state path for the session, from which we get the sequence of computed behaviors. This is a classical step for standard HMMs, we explain below how it works for MS-HMMs.
To segment a session into elementary behaviors, one builds three large HMMs λ_{rd}, λ_{rs}, λ_n by concatenating all HMMs corresponding respectively to reading features, resources features and navigation features. The global MS-HMM model denoted λ is built from these three asynchronous models, by imposing synchronization points at each leaving state, i.e. the three models are forced to leave a behavior model at the same time in each stream. The likelihood of a session is given by:

$$P\left(o_1^T / \lambda\right) = \sum_{S_{rd}, S_{rs}, S_n} P\left(rd_1^T / S_{rd}, \lambda_{rd}\right) P\left(rs_1^T / S_{rs}, \lambda_{rs}\right) P\left(n_1^T / S_n, \lambda_n\right) P\left(S_{rd}, S_{rs}, S_n / \lambda\right) \quad \textbf{(6)}$$

where S_{rd}, S_{rs}, S_n are the paths in λ_{rd}, λ_{rs}, λ_n. The synchronization consists in setting $P\left(S_{rd}, S_{rs}, S_n / \lambda\right)$ to 0 if the constraint is not verified. Otherwise, $P\left(S_{rd}, S_{rs}, S_n / \lambda\right)$ is set equal to $P\left(S_{rd} / \lambda_{rd}\right) . P\left(S_{rs} / \lambda_{rs}\right) . P\left(S_n / \lambda_n\right)$.

4 Experiments

We describe now two series of experiments. In a first series we want to categorize whole user sessions. Remember that sessions were built with one typical behavior in mind, so that each session should correspond to one class. In a real situation, this corresponds to the case where a user is supposed to have the same behavior for an entire session. In a second series of experiments, we want to track the behavior of the user and detect its behavior changes. This amounts to segment user sessions into reference behaviors. This is a more realistic situation for most hypermedia users. For our experiments, we concatenated all the elementary user sessions in the database using a random ordering, producing large sessions where the user changes behavior. All the evaluations have been performed using a 26-fold cross-validation.

It must be noticed that, even in a closed and controlled environment like the one we are dealing with, user behavior classification is difficult and has intrinsic limitations. Even with a clear goal in mind, a user goes back and forth between different strategies during a session, which makes difficult an accurate classification. The elementary behaviors we are using are only rough abstract representations of the potential user behavior.

Since we built the database using predefined scenario, we know the label of each elementary session. It is thus possible to perform supervised learning for both classification and segmentation. We performed supervised learning with both HMMs and MS-HMMs. Although this could make sense for user behavior classification in some controlled environments, it is usually more realistic to consider the problem as an unsupervised learning problem where sessions are unlabeled, and the goal is to identify typical user behaviors from scratch. We thus performed experiments with unsupervised learning with HMMs and MS-HMMs. The interpretation of the discovered behaviors is complex and the evaluation of unsupervised methods is an open problem. We thus provide below performances of unsupervised methods with regard to the known labels of elementary sessions. Although this is not really satisfying, this provides interesting hints for measuring the ability of these methods to detect user behaviors. Note that performances obtained using supervised methods provide an upper bound of the performances that could be obtained for session classification and segmentation.

4.1 Session categorization

Here whole sessions have to be classified according to an underlying behavior. We used two evaluation criterions, the standard *correct classification* (*CC*) percentage, and a *weighted accuracy* (*WA*) criterion where confusions between classes have different weights. The idea behind *WA* is that confusions between behaviors do not all have the same importance since user help actions for some classes may be very similar. In our *WA*, confusions between *Scanning* and *Exploring* and between *Searching* and *Wandering* are set equal to ½ while all other confusions are set equal to 1.

For supervised learning, we trained 4 models, one for each typical behavior. A standard HMM model working on whole frames, has 7 states. A MS-HMM model consists of 3 HMMs, one per feature subset, with 3 states. The number of states in the models have been fixed using cross validation.

For behavior clustering (unsupervised learning), we first determined an "optimal" number of clusters using the F-statistic, which is a cluster homogeneity measure. We found an optimal number of 6 clusters. We then learned a mixture of 6 models. Training sessions were then clustered according to the model with greatest likelihood. After training, each cluster has been labeled into one of the 4 classes according to the majority of labels it contains. *CC* and *WA* criteria may then be computed.

Table 1 sums up our results. Both *CC* and *WA* are reasonably high for supervised models: elementary behaviors can be recognized rather accurately from low-level navigation data. As may be seen, Markovian models perform similarly to MLP although these models do not capture the same kind of information, this shows that the dynamic information is partially handled using Markovian models. Although *CC* and *WA* are noticeably lower for unsupervised training, it can be seen that a reasonable proportion of the sessions is correctly classified. This shows that unsupervised classification on user traces allows capturing valuable information on the user behavior. This also shows the difficulty of this task. Going further in the evaluation of unsupervised systems would necessitate a manual analysis of the clusters, this is beyond the scope of this paper.

Table 1. Behavior classification accuracy and weighted accuracy for 3 supervised systems (HMMs, MS-HMMs, MLP) and 2 unsupervised systems (HMMs and MS-HMMs).

Training mode	System	HMM	MS-HMM	MLP
Supervised	*Correct classification*	79	76	74
Supervised	*Weighted accuracy*	85	84	83
Unsupervised	*Correct classification*	69	65	-
Unsupervised	*Weighted accuracy*	78	76	-

4.2 Session segmentation

Here, a system has to detect the user behavior changes in a long session, and to recognize these behaviors. A segmentation system receives as input a sequence of frames and outputs a sequence of labels, one for each frame. In our controlled experimental setting, this computed sequence has to be close to the actual label sequence. Different measures have been proposed for comparing discrete sequences, we use here the *edit distance* between computed and desired label sequences [1]. This is a classical measure which computes *insertions*, *deletions* and *substitutions* between the two strings. The *correct recognition* percentage is then 1 minus substitution and deletion percentages. Note that this does not take into account the duration of each detected behavior. We made this choice considering that it was not important to detect the exact time where the user changes his exploration strategy, but rather to detect the change of strategy within a reasonable delay. The Edit distance reflects this idea up to a certain extent.

Both for supervised and unsupervised settings, models are first trained on elementary sessions as for classification. Each model is then associated to one of the 4 predefined classes. Models are then used to segment a large session where elementary sessions have been concatenated. The computed sequence is compared to the desired sequence via the Edit distance. Table 2 shows the experimental results.

Again performances of supervised models are satisfying and only show a small drop compared to the simpler task of classification. MS-HMMs are still 4% higher than simple HMMs. As for categorization, Markovian models perform similarly to MLP. On the other hand, performances of unsupervised systems are 40 % below the supervised upper bound. The lower classification ability carries over segmentation. Also, it must be noticed once again that unsupervised systems are evaluated using supervised labels so that a mismatch between discovered and labelled behaviours lead to poor results and may not reflect the eventual relevance of unsupervised systems.

Table 2. Edit-distance rates between correct and predicted behavior sequences, with substitution cost =1 and deletion cost = insertion cost = 2, for 2 supervised systems and 2 unsupervised systems (standard HMMs and MS-HMMs in both cases + MLP for supervised mode only).

Training mode	Edit-distance	% Correct	% Susbt.	% Del	% Ins
Supervised	HMM	78	14	9	12
Supervised	MS-HMM	75	16	10	10
Supervised	MLP	73	16	11	13
Unsupervised	HMM	35	55	10	14
Unsupervised	MS-HMM	39	50	11	13

5. Conclusion

We proposed a series of new features for categorizing user navigation patterns in rich hypermedia systems. We then developed two sequence models for the classification and tracking of user behavior. Experiments were performed on a representative hypermedia system using a controlled navigation database. Results show that whole session classification performs pretty well even in an unsupervised setting, on the other hand, behavior tracking is more difficult when there is no a priori knowledge of what a typical session is. More work is then needed to limit the drop in performance between classification and segmentation. Still, these are encouraging results for the automatically assisted development of hypermedia user-centric help systems.

Acknowledgment : This project is realized as part of the RNTL project Gicsweb funded by the French Ministry of Industry.

Bibliography

1. Atallah, M.J. (ed.), Algorithms and Theory of Computation Handbook, CRC Press LLC, 99.
2. Brusilovsky P., Adaptive Hypermédia, *User Modeling and User-Adapted Interaction*, 2001.
3. Cadez I., Gaffney S., Smyth P., A general probabilistic framework for clustering individuals and objects, *In Proceedings of the Sixth ACM International Conference on Knowledge Discovery and Data Mining*, 2000.
4. Canter D., Rivers R., Storrs G., Characterizing User Navigation through Complex Data Structure, *Behavior and Information Technology*, vol. 4, 1985.
5. Catledge L., Pitkow J., Characterizing Browsing Strategies in the World Wide Web, *Computer Networks and ISDN Systems*, 1995, vol.27, No.6.
6. Dupont S. and Luettin., Using the Multi-Stream Approach for Continuous Audio-Visual Speech Recognition: Experiments on the M2VTS Database, *Int. Conf. on Spoken Language Processing*, 1998.
7. Gauthier N., Artières T., Dorizzi B., Gallinari P., Strategies for combining on-line and off-line informations in a on-line handwriting recognition system, *Int. Conf. Document Analysis and Recognition,* 2001.
8. Horvitz E., Breese J., Heckerman D., Hovel D., Rommelse K., The Lumière Project: Bayesian user modeling for inferring the goals and needs of software users, *UAI 98*.
9. Varga A., Moore R., Hidden Markov Model decomposition of speech and noise, *International Conference on Acoustics, Speech and Signal Processing*, 1990.
10. Zukerman I., Albrecht D., Nicholson A., Predicting users' requests on the WWW, *User Modeling*, 1999.