Statistical machine learning for tracking hypermedia user behavior

S. Bidel, L. Lemoine, F. Piat, T. Artières, P. Gallinari

LIP6, Université Paris 6
8 rue du capitaine Scott, 75015, Paris, France
{Sylvain.Bidel, Laurent.Lemoine, Frederic.Piat,
Thierry.Artieres, Patrick.Gallinari}@lip6.fr

Abstract. We consider in this paper the classification and tracking of user navigation patterns for closed world hypermedia. We use a number of statistical machine learning models and compare them on different instances of the classification/tracking problem using a home made access log database. We conclude on the potential and limitations of these methods for user behavior identification and tracking.

1 Introduction

The development and complexity increase of hypermedia systems accessible by a large variety of users has created a need for developing tools to help the user meet his information need. For such applications where navigation plays an important role, it is generally useful to characterize dynamically the user behavior. On line positioning of individuals inside a user taxonomy is an important information for defining help strategies. Examples of such characterizations are the classification of user behavior among different pre-defined categories, the discovering of behavior changes or the tracking of user behavior. This type of analysis is a preliminary step for the development of adaptive help systems and it is usually performed by analyzing on line user traces.

We consider here how machine learning techniques could be useful for characterizing dynamically the behavior of a user, navigating a closed world - rich information content hypermedia. In this context, we focus on the automatic discovery of user navigation behavior from the low-level information provided by the temporal sequences of navigation actions (visited document, click, scroll, etc). The goal is to follow the user during its navigation and to track its navigation behavior. For that we need to consider the sequences of user interactions with the hypermedia. We first propose a number of features which allow to characterize the sequence of user actions with respect to the user typology, and which are adapted to the rich content of the hypermedia. For tracking the user, we investigate a number of statistical machine learning models for dealing with temporal data: a neural network, Markovian models (Markov Chains and Hidden Markov Models), we also introduce Multi-Stream Markov Models which allow to process simultaneously different feature sequences occurring at different time scales and asynchronously. For comparing and analyzing

the models we propose two series of experiments, we first evaluate the models for the classification of homogeneous user sequences, we then consider the more difficult – and more interesting- problem of user behavior tracking. For these two tasks we take two approaches to the learning problem: supervised and unsupervised learning. They correspond to two different strategies and needs for developing user models. Supervised learning might be adequate for a fixed number of user categories with well defined user behaviors, unsupervised learning makes easier the incorporation of new categories and the development of adaptive systems able to incorporate new or evolving populations.

Note that, 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.

The evaluations are performed on a home made basis for which 26 users have been enrolled for a total of about 16 navigation hours of a multimedia encyclopedia. Besides comparing the different models for the classification and tracking tasks, we also discuss the potential and limitations of automatic tools for analyzing user action sequences.

The paper is organized as follows. We first introduce in §2 a navigation patterns typology. Then, we describe in §3 the database used in our experiments. In §4, we describe our behavior models and present the supervised and unsupervised strategies. Then, we give experimental results in §5.

2   High level navigation strategies

Although most researchers distinguish broad user navigation strategies (e.g. browsing and searching), there is no general agreement on a typology. Defining a clear classification between different behaviors is also made difficult since strategies are not mutually exclusive and users frequently go back and forth between them – e.g. browsing may be used to achieve searching, etc. For categorizing the behaviors, we use here a popular taxonomy by Canter [4], it distinguishes the following high level behaviors:

- *Scanning*: seeking an overview of a theme (or of connected themes) 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 fashion without particular goal or strategy.

3   Database and Feature Extraction

For this work, we used « The XXth century encyclopedia », initially a cultural CD-ROM[1] reconfigured as an Internet site. This is a typical "cultural" hypermedia system,

---

[1] Distributed by Montparnasse Multimédia company.

it contains about 2000 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 homogeneous user data sessions in a controlled fashion, by asking 26 users to fill out questionnaires by navigating through the encyclopedia. The questions for each session were chosen in order to induce a given navigation behavior according to the typology in §2. For instance, a question asks the user to extract some important dates from a particular theme. This prompts the user to view several pages of this theme without having to read them thoroughly, which corresponds to the "Scanning" behaviour. For the "Exploring" behaviour, the user was asked to fully read a few articles. For the "Searching" and "Wandering" Behaviors, the users were asked to retrieve from the whole encyclopaedia a particular picture (for Searching) and to pick any one they liked (for Wandering). The sequences of user actions (traces) recorded by the navigator and associated to each questionnaire is called an *homogeneous user data session*. Each session is then labelled by the corresponding high-level behaviour. These labels will be used for the evaluation and for training supervised classifiers.

104 data sessions were thus gathered, 26 for each of the 4 behaviors. Navigation data are sequences of dated events (page access, click, scroll, query on the search engine, etc) which are collected all along the user session. These traces are then processed to compute sequences of feature vectors or frames. A frame was computed about every minute and overall, this yields over 900 frames. We investigated many different features for characterizing user behavior 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. Using reference reading times for each paragraph we compute reading rates for the first quarter and for the rest of the document (applies when a document is 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 type of resources used. They may be articles (the real content of the hypermedia, the leaf 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 as features the percentage of time spent on these three kinds of resources.
- The "*navigation*" subset characterizes the navigation focus, it indicates whether the user is focused on one theme or spread onto several. Let the similarity of two themes be the cosine between the two vectors representing the themes key-words. We define two navigation features. The first one is the average distance between the themes of successive pages accessed during the frame duration. The second feature does not take into account the visiting order of the pages but measures the global variability of themes visited. For that, we first determine the main (focus) theme as the one whose average distance to all visited articles is minimal. We then compute

the average distance of the visited themes to this focus, weighted by the time spent on each theme.

4 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 $o_1^T=(o_1, ..., o_T)$ denote a sequence of T frames, o_t being the tth frame in the sequence. We want to identify different user behaviors, to do so, we trained models of frame sequences. Such a model, B, allow computing sequence likelihood $P(o_1^T/B)$. We investigated various statistical machine learning models, Multi-Layer Perceptrons (MLPs), Markov Chains (MCs), Hidden Markov Models (HMMs) and Multi-Stream variations of Markovian models.

We used MLPs since they are known to be efficient for discrimination tasks. A 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:

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

We also used Markovian models since they have shown strong abilities for various signal and sequence modelling and classification tasks. We use one Markovian model per behavior (either MC or HMM), with an ergodic topology (any transition allowed), and diagonal covariance Gaussian densities in the case of HMMs. Learning and recognition algorithms are classical ones and are not detailed here.

The underlying hypothesis for Markovian models is that the modeled process is locally stationary and a transition in the Markov model corresponds to a skip from one of its stationary states to another one. A consequence is that all features in the frames are assumed to obey a synchronous process. This assumption does not correspond to the features used here which do not change synchronously. Hence, we propose to use a variant of Markovian models called Multi-Stream Markovian models [9]. We briefly present below the principle of multi-stream HMMs (MS-HMMs), the case of MS-MCs is similar.

MS-HMMs allow combining multiple partially synchronous information streams or modalities [6, 7]. In our study, a MS-HMM is a combination of three stream-HMMs each operating on a different information stream corresponding respectively to *Reading*, *Ressources* and *Navigation* frame sequences (see §3). 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. We have chosen 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 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:

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

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 section 4.3.

4.1   Supervised and unsupervised learning

For our experiments, we investigated supervised and unsupervised learning. Both have been performed using the assumption that a homogeneous user data session in the database (as defined in section 3) corresponds to a user who doesn't change his behavior all along this session.

For supervised learning, we used the labelling of our database into the four elementary behaviors as described in §2. For Markovian models, we learned four models (one for each behavior), each behavior model is trained to maximize the likelihood of associated training sessions. For the MLP, one MLP is trained to discriminate between the frames from the four behaviors, using a Mean Squared Error criterion.

We also investigated unsupervised learning for Markovian models (we did not performed unsupervised experiments with MLPs since this model is not well adapted to this task). To do this, we consider a mixture of N probabilistic behavior models. The probability of a sequence is given by the following mixture of sequence models:

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

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. Learning consists in maximizing the likelihood of all training sessions given this mixture model. Since for unsupervised learning we do not know which behavior a training session belongs to, we use an EM procedure where missing data are posterior probabilities of behavior $P\left(B_i / o_1^T\right)$. This algorithm performs a clustering of user sequences. Here is the sketch of our clustering 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.

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

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:

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

4.2 Behavior categorization

In a first step we have compared the different methods on the classification of homogeneous user data sessions (§3). For each model, MLP, MCs, MS-MCs, HMMs or MS-HMMs: the sessions are classified according to the model maximizing the sequence likelihood ((1) was used for MLP).

4.3 Behavior tracking

Usually sessions are not homogeneous and exhibit multiple successive behaviors, the goal is then to track on line the user behavior. In this case, we make use of global session models built from elementary homogeneous models. For Markov models, a global Markov model is built by concatenating the leaving state of each behavior model to the entering state of each behavior model. Then, considering an unknown session, a dynamic programming algorithm finds the optimal state path for the session, from which we derive the most likely sequence of typical behaviors. This corresponds to the segmentation step for standard Markovian models (MCs and HMMs).

For MLPs, a similar scheme may be used. Let us explain below how it works for MS-HMMs (it is similar for MS-MCs).

To segment a session into elementary behaviors using MS-HMMs, one builds three large HMMs λ_{rd}, λ_{rs}, λ_n by concatenating, as above, 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 paths in each stream are forced to leave their behavior model at the same time. The likelihood of a session is given by:

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

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 $P\left(S_{rd} / \lambda_{rd}\right) . P\left(S_{rs} / \lambda_{rs}\right) . P\left(S_n / \lambda_n\right)$.

5 Experiments

We now describe the two series of experiments. In the first series we categorize homogeneous user sessions. This does not usually correspond to a realistic scenario, but it allows to perform a preliminary evaluation on a simplified task. In the second series of experiments, we want to track the user behavior and detect its behavior changes. This amounts to segment user sessions into reference behaviors. This is a more realistic situation. For this second series of experiments, we concatenated all the homogeneous user sessions in the database using a random ordering, producing large sessions where the user behavior changes.

All the evaluations have been performed using a 26-fold cross-validation. Each experiment consists in training the system using all user data but one, and to test on the remaining user data.

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 of sessions. The elementary behaviors we are using are only rough abstract representations of the potential user behavior.

Since we built the database using a 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 all the models described in §4. Although this strategy could make sense for user behavior classification in some controlled environments, it is 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 unsupervised learning experiments with Markovian models only since MLP is not well adapted to unsupervised learning. 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 (i.e. supervised) labels of elementary sessions. Although this is not fully 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.

5.1 Session categorization

Here whole sessions have to be classified according to an underlying behavior. For evaluating the models, we used two 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 respectively weighted by a ½ factor, all other confusions are assigned a weight equal to 1, these weights have been fixed by hand.

For supervised learning with Markovian models, we trained 4 models, one for each typical behavior. A standard HMM (MC) model working on whole frames, has 7 states. A multi-stream HMM (MC) model consists of 3 HMMs (MCs), one per feature subset, with 3 states. The number of states in the models has been fixed using cross validation. We use one MLP that is trained in discrimination mode.

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 most supervised models: elementary behaviors can be recognized rather accurately from low-level navigation data. Behavior may be recognized with up to 65% accuracy using only one frame (1 minute), and with up to 79% for whole sessions (about 5' in average). As may be seen, HMM models and MLP perform similarly, and outperform MC models.

Although *CC* and *WA* are noticeably lower for unsupervised training, it can be seen that a reasonable proportion of the sessions is again 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.

For supervised learning all three models HMM, MS-HMM, MLP, do perform similarly. The MS-HMM is unable to take benefit here from its better ability to model the sequence data. For unsupervised learning the same conclusion holds (MLPs being left out).

Table 1. Correct classification percentage (CC) and weighted accuracy (WA) for behavior classification task for supervised and unsupervised systems.

Training mode	System Criterion	HMM	MS-HMM	MC	MS-MC	MLP
Supervised	*Session CC*	79	76	57	63	74
Supervised	*Session WA*	85	84	67	72	83
Unsupervised	*Session CC*	69	65	61	61	-
Unsupervised	*Session WA*	78	76	70	70	-

5.2 User behavior tracking

Here, the system has to detect in a long session the behavior changes, 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 have been using 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.

For supervised learning, models are first trained on elementary sessions as for classification. For unsupervised learning, the class of each homogeneous session is supposed unknown. 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.

Table 2. Edit-distance rates between correct and predicted behavior sequences, with substitution cost =1 and deletion cost = insertion cost = 2, for supervised and unsupervised systems.

Training mode	Edit-distance	% Correct	% Susbt.	% Del	% Ins
Supervised	HMM	78	14	9	12
Supervised	MS-HMM	75	16	10	10
Supervised	MC	49	38	13	14
Supervised	MS-MC	61	29	10	17
Supervised	MLP	73	16	11	13
Unsupervised	HMM	35	55	10	14
Unsupervised	MS-HMM	39	50	11	13
Unsupervised	MC	37.5	51	12	12
Unsupervised	MS-MC	39	44	14	12.5

Again performances of supervised models are satisfying and only show a small drop compared to the simpler task of classification. MLP, HMMs and MS-HMMs are still higher than MCs and MS-MCs. This is an encouraging result since it shows the feasibility of behavior tracking. On the other hand, performances of unsupervised systems drop 30 % below the supervised upper bound for all models. The lower classification ability carries over segmentation. It looks like tracking is not possible in an unsupervised setting. Note however, that this evaluation of unsupervised systems for segmentation is even more questionable than for categorization since there is no clear frontier between the different categories. An analysis of the segments inside each cluster should be performed in order to assess the relevance of these models.

Globally, these controlled experiments show that classical machine learning tools like HMMs and MLPs operating on adequate navigation features allow extracting significant information from on line user information. These models behave similarly and more sophisticated models did not bring any improvement. All models allow to operate on line on the sequences of user actions. Both classification and tracking do perform at a reasonable level in a supervised setting, although the performances shown here could be an upper bound for this type of system. For unsupervised learning which probably corresponds to the more interesting scenario for analyzing user actions, things are more complicated and e.g. nothing could be definitely concluded

from the results of the tracking experiments. Further investigations and interpretation of the data segmentation are needed to go further. However it seems clear that in this setting, additional information like e.g. user interaction is needed in order to confirm or not the model decisions. In both cases, generative models like HMMs allow to incorporate new type of user behaviors and this is an advantage compared to discriminate methods like MLPs.

6   Conclusion

We proposed a series of new features and investigated various statistical machine learning models for the categorization and tracking of user navigation behaviors in rich hypermedia systems. Experiments were performed on a real hypermedia system using a controlled navigation database. Results show that session classification and tracking performs well in a supervised setting, but that performance drops for unsupervised tracking. It is not clear yet whether this is an intrinsic limit of the unsupervised approach to tracking or a side effect of the evaluation criteria. In all cases it seems that additional information from the user must be taken into account if we want reliable tracking and classification.

Bibliography

1. Atallah, M.J. (ed.), Algorithms and Theory of Computation Handbook, CRC Press LLC - 1999
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.W., Nicholson A.E., Predicting users' requests on the WWW, *UM 99*.