Describing user's search behaviour with Hidden Markov Models

Sebastian Dungs

University of Duisburg-Essen, 47057 Duisburg , Germany,
dungs@is.inf.uni-due.de,
http://www.is.inf.uni-due.de

Abstract. This paper introduces Hidden Markov Models (HMM) for modelling user's search behaviour in a digital library search scenario. Three potential applications of these models have been identified in the literature: System evaluation, simulation of user activity and user guidance. Models have been generated from 36 eye tracking log files, where smaller models up to five hidden states provided best prediction quality to complexity ratio. While in its current state this work makes simplifications limiting its expressiveness, many possible improvements and their benefits are presented. Future work will include a larger observation vocabulary e.g. by combining eye tracking and systems logs as well as exploring other potential applications of HMM like semantic clustering of rich log data and discovery of human cognition pattern in (digital library) search sessions.

Keywords: Digital Libraries, Hidden Markov Models, User Modelling, Interactive Information Retrieval, User Guidance

1 Introduction

The Interactive Probability Ranking Principle (IPRP) [7], a theoretical framework of Interactive Information Retrieval (IIR), was introduced by Fuhr to provide a foundation for the functional design of modern IIR systems. While the framework does not specify how to estimate its parameters in practice, it was later used by Tran and Fuhr [17] to develop a method for the analysis of IIR that utilizes Markov Models (MM). Given data from a lab experiment with fixed book-finding tasks, the authors estimated the time until users retrieve the next relevant document by examining user's previous actions. They proceed to discuss various applications of these calculations. Proposed ideas include generation of artificial user activity to simulate the effects of system changes without actually building them by tuning model parameters as well as adding user guidance algorithms to search systems that follow IPRP principles.

In Tran and Fuhr [17, 16] each possible user action is modelled explicitly in a separate state. This leads to models that either oversimplify reality by artificially reducing user's choices or are overly complex with a considerable increase in states. Models with many states also require substantially more observation

data for estimating model parameters which might not be easy to acquire for researchers [17].

Hidden Markov Models (HMM) extend plain Markov Models by introducing *hidden states* that represent the unknown components of an agent, i.e. the user or a system. All states have associated transition probabilities analogue to MM. HMM also include the concept of *observations*, which occur with every state change with their respective probabilities. The set of all possible observations is called the *observation alphabet* and comprises all aspects of the target domain that can be observed by the researcher, e.g. different entry types in a system log. Hidden states as an embedded stochastic process allow to overcome the major limitation of the usage of MM in the domain of IIR modelling—overproportional inflation of state space with growing problem complexity.

In the course of this dissertation, the general ideas of user guidance according to IPRP will be directed towards application in digital library context since IIR is often not sufficiently supported by current productive digital libraries. Furthermore, this dissertation will provide researchers with tools to evaluate system changes quickly and economically by means of user simulation. Eventually, providing fundamental research for including advanced supporting measures in digital libraries will lead to systems that are more effective and efficient to use.

This paper describes the current state of work in that direction—specifically how HMM can be used to quantitatively describe the process of searching in a digital library. The focus lies on an abstract representation of mental states of users that cannot be observed directly. While HMM have been used extensively in other domains like speech recognition, there is not much research on how to represent human cognition in IR using HMM. Nevertheless, this method has strong potential to be beneficial in interactive information retrieval research since user actions are modelled as observations in HMM. Therefore, model size is not directly determined by the set of possible actions making this approach more general compared to [17]. However, this advantage comes with the downside of increased interpretation and computation complexity. The latter can be overcome by using approximating algorithms for model generation. Qualitative interpretation, however, is a key challenge in dealing with HMM and will to be subject of ongoing research.

The work by Tran and Fuhr led to three hypothesis, which can be evaluated by the proposed research ideas presented in this paper: First, building quantitative models of search behaviour will lead to a deeper understanding of the search process, especially if combined with existing and well established qualitative models. Furthermore, application of HMM will allow to predict users' future actions and enable researchers to build systems with support for higher search activities according the IPRP. And finally, if built accordingly HMM can also be used to interpret rich system logs, i.e. logs that capture many different types of user interaction with the system. The output of the proposed process in this case would be a model containing a set of states, each of which has a set of possible observations—the user actions—including their probabilities. observations with

a high probability in the same hidden state are likely to be related to the same mental state of the user and could therefore be aggregated.

Based on the same eye tracking logs used by Tran and Fuhr [16], several models of different complexity—with respect to quantity of hidden states—have been built. The goal was to demonstrate the applicability of HMM in the context of this particular scenario of search in a digital library. The hypothesis is that similar to [17] models based on limited test samples can then used to derive general probabilistic calculation rules of expected task completion times and success rates.

Apart from cognitive sense making, in the context of search behaviour modelling HMM can be used to make predictions about users' future actions by examining current interaction data. Based on these predictions an IIR system can make suggestions to support the user in achieving her goals. This has potential to help in overcoming a main gap in IIR research: Developing IR systems that offer support while users are engaged in higher level, complex search tasks. Given Bates' level of search activities [4], current systems are limited to supporting the two lower levels of moves and tactics. However, *strategems* [3] are not supported by popular search engines leaving room for input by the research community. The need to focus on supporting higher search activities is even more prominent when dealing with complex information needs, i.e. in digital library research.

The remainder of this paper is structured as follows. Related work in the area of modelling search behaviour and successful application of (H)MM in user modelling is described in Section 2. In Section 3 first the data set used in this work is depicted. Additionally, the process of model generation is presented. Section 4 describes preliminary results and Section 5 covers the future work and paints a vision of how built models can be used in real time applications.

2 Related work

Besides the work of Fuhr and Tran mentioned above, Markov Models have been applied in various fields of research and practical application, like financial modelling [10, 18] or computational biology [12, 15].

Several variations and extensions of Markov models have been used in the literature. Liu et al. [13] for example used *Continuous-Time Markov Processes* for modelling users' browsing behaviour in web search. *Partially Observable Markov Models* (POMM) were used by Wang et al. [19] and He et al. [9] taking into account the observable click data to estimate searchers' viewing behaviour. By using POMM the authors relaxed the assumption that all state changes lead to observations present in the log data. Yue et al. [20] used Hidden Markov Models to model collaborative, exploratory search sessions. The authors found HMM to be suitable for automatic analysis of search processes. Luo et al. [14] also considered the addition of a reward function in modelling search sessions by using *Partially Observable Markov Decision Processes* (POMDP).

Searchers' interaction extracted from logs has been used in the past to predict search success by Hassan et al. [8]. Ageev et al. [1] were able to reproduce these findings using their own data. Furthermore, the authors used *Conditional Random Fields* to perform the same task with superior performance.

Azzopardi et al. introduced *Search Economic Theory* (SET) [2]. It is based on the IPRP framework [7] and estimates users' effort during search using a cost function. The main contribution of this work was to compare different search strategies regarding their cost of interaction to expected gain ratio using simulated user interactions.

3 Building Hidden Markov Models

Preliminary results presented in this paper are based on user experiment data gathered and described by Fuhr and Tran [16]. This data consists of two separate logs of 12 participants carrying out 3 tasks each, searching for books in a crawl of the Amazon.com book collection. One log was generated by the search system used in the experiment and covers all events that can directly be observed by the system, like the execution of a search query or storage of a result in the basket tool. All events are associated with a time stamp. The other log contains eye tracking data on *fixation* level.

For the purpose of this paper we define a fixation as a glimpse at a specific location on screen with a minimum duration of 80ms. Each fixation has a time stamp as well as a duration and is associated with one of four pre-defined *Areas of Interest* (AOI) which are derived from the interface's four[1] main functional areas shown in Figure 1:

- *Query*: covering the query input boxes (orange & blue)
- *Result*: the result list and filter options (red & cyan)
- *Details*: a detailed view of one specific document in the result set (purple)
- *Basket*: a storage space designated for relevant documents (grey)

3.1 Data preparation

In the current state of this work, Hidden Markov Models were built from eye tracking logs only. The reason for this is to reduce complexity during the first steps of research. Each distinct value or event present in the logs increases the vocabulary size of the observation sequences. While larger vocabulary size does not necessarily lead to more complex models regarding the amount of hidden states, it still makes models harder to interpret. Limiting model generation to a small observation vocabulary allows models to still be visually comprehensible and printable and considerably decreases computation times.

36 observation sequences of length up to 3823 fixations can be created using the data provided by Fuhr and Tran [16]. Given the AOI introduced above, the

[1] Not all depicted AOI were used in present work

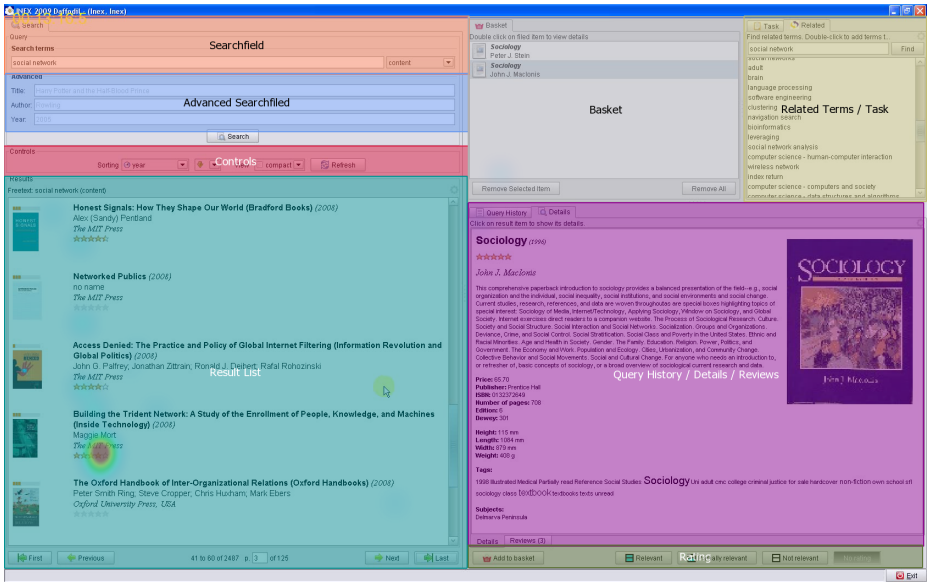

Fig. 1. Interface used in experiment. AOI are shown as coloured rectangles.

observation alphabet has a size of four. Figure 2 shows a timeline of user's viewing behaviour (x-axis given in ms) mapped to these four areas (y-axis). Currently not all fixations were used as an input during model generation. For reasons of feasibility only transitions between AOI—changes on y-axis—were considered to be an observation while retaining the ordering. Length of fixation sequences on a single AOI as well as length of the fixation itself are currently not considered as an input signal in the model but can be added in the future.

This preprocessing lead to 36 sequences of observations (S_i) of length from 37 up to 220. The sequences were grouped by experiment participant and used accordingly during model learning stage which is described in the following paragraph.

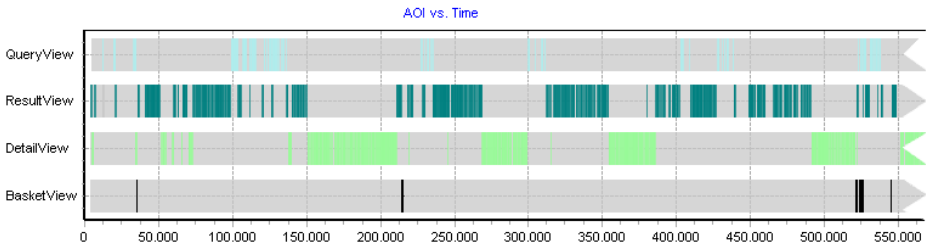

Fig. 2. Visualization of an AOI log file

3.2 Model generation

As described by Chau et al. [5] computationally feasible algorithms for solving HMM discover local optima depending on start value selection for transition and observation probabilities. Therefore, only approximate solutions can be given. For the scope of this work, a set of 10.000 initial value distributions has been created randomly. Given the 12 sets of observation sequences, a leave-one-out cross-validation procedure has been used to train each of the initial models using Jahmm's implementation of the Baum-Welch algorithm[2]. After the learning algorithm converged, the remaining sequences' probabilities have been calculated using the forward algorithm. The initial model with the lowest average prediction error was kept for further analysis. Note that the best suitable start value distribution has finally been used to build a model using all available sequences.

Apart for start value distribution, another general challenge encountered while generating HMM is choosing model's size appropriately. There is no known algorithm to compute the correct hidden state count (n) beforehand. Therefore, the above described process was repeated for different n from 2 up to 10. From each of these runs only the best model was persisted. Note that each run uses different sets of 10.000 random start value distributions since these already determine the model size in terms of hidden state count.

3.3 Descriptive strength of models

As was emphasized, there is no known a-priori solution for finding the correct model size. Instead, different models have be generated and evaluated regarding their descriptive strength. The general assumption is that models become more descriptive with increased size while obviously also becoming more complicated to interpret. Furthermore, larger models may overfit to training data. To develop a rough understanding on reasonable model sizes the observation sequence probabilities $P(S_i)$ can be used. Details of the calculated errors for different model sizes' respective best model are given in Table 1. Note that because of the naturally overall very low probabilities a logarithmic scale has been used, i.e. higher values indicate better models.

Looking at the data gathered by learning models, two interesting observations can be made: First, a *saturation point* can be observed at a model size of five hidden states, where the rate of prediction error reduction by introducing additional states slows down considerably. Furthermore, adding a ninth state even increases model's predicted error over the 8-state version. This can be interpreted as a strong sign of overfitting. Consequently, very large models have been excluded from further evaluation and future calculation dealing with this particular data set will focus only on models up to five hidden states as a trade-off between quality and complexity of models.

[2] https://github.com/KommuSoft/jahmm

Table 1. Estimated error for different state counts

State count	$log(P(S_i))$
2	-116.989
3	-110.645
4	-107.465
5	-102.516
6	-101.565
7	-100.477
8	-99.109
9	-100.291
10	-99.581

4 Preliminary results

In this section preliminary results generated by the procedure described in the previous paragraphs are presented. One major problem while visually analysing HMM is that of labelling the hidden states. Furthermore, graphs quickly become confusing if many potential observations are shown. For easier comprehension consideration will therefore focus on the two smallest models created.

The best HMM for the two state case is shown in Figure 3. For an n-state HMM the states are depicted as circles labelled S_0 to S_{n-1} with their transitions drawn as solid lines including probabilities. Observations have unique labels (according to AOI) and are drawn as rectangles with dashed lines that indicate probabilities. Going through the details of this graph, a few things are striking: Primarily, S_1 most likely represents the user going through the result list because this state has a .86 probability of emitting a *Result* observation. S_0 is more ambiguous since all other possible actions are unified in this state. The observation probabilities are more evenly distributed across three of the four categories. Nevertheless, this state never emits *Result*, clearly separating it form S_0. It can be observed by examining the values that on average a *Basket* observation is less likely to appear than *Query* or *Detail*. Specifically, it takes the user on average almost two queries and more than three closer document examinations to find and store a relevant document.

In the two state case S_0 is going to be the starting state with a probability of .79. Given previously mentioned state interpretation, a much higher value close to 1 would have been expected. The reason for this is believed to be originated in noise in the data. Currently all observations are considered based on the above stated pre-processing algorithm. However, the experiment set-up featured an unfamiliar user interface much more complex than common web search interfaces. It is reasonable to assume that users might have been overwhelmed at the beginning of the experiment runs. The first few sections of fixations could be attributed to an orientation phase which was then followed by the actual execution of the tasks.

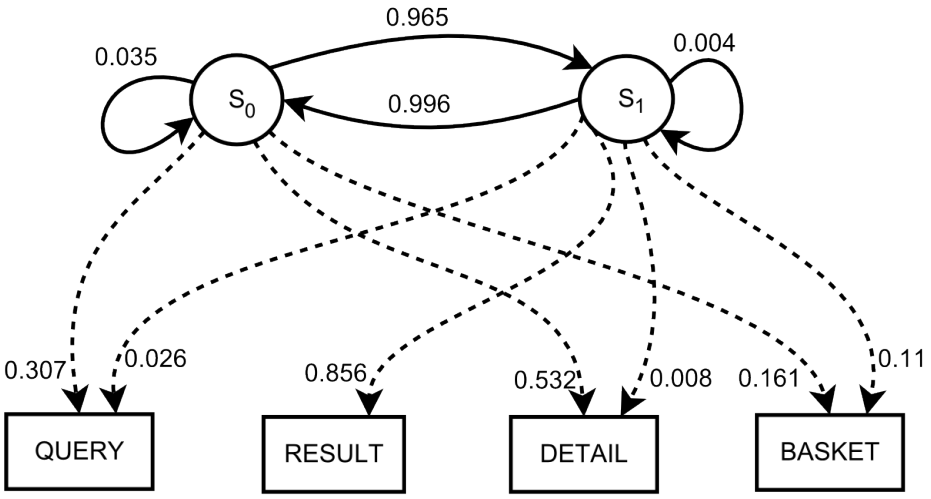

Fig. 3. The two hidden state model

Summarising the observations it can be said that a two state HMM divides user actions in reading result snippets on the one and all other search activities on the other hand. It has to be noted that—due to the chosen data preparation method—the models' looping probabilities are strongly biased towards very low values. Therefore, a *Result* observation has to be interpreted as a user working through a result list as a whole, not looking at one particular entry in the list.

The three state model—not shown because of space limitations—shares some properties of the two state version, i.e. S_0 that is most likely to emit only *Result* (.91). However, with the introduction of the third state comes a clearer differentiation between query and document related user actions. S_2 has a .77 probability of emitting *Details*, whereas S_1 has almost equal probabilities for *Query* (.45) and *Basket* (.54). The general low likelihood to enter S_1 is in line with users' behaviour—issuing few queries as well as storing few documents per task. An .08 probability of a *Basket* observation in S_0 would represent users that mark a document as relevant without prior inspection of the document details based on just the snippet and is in line with observations in user experiments.

The start state probability distribution does not show a clear indication of a starting state with S_0 and S_2 being almost equal in probability (.42 vs. .44). Again, these values have to be reexamined after further cleaning of the data has been performed.

The visual interpretation of larger models quickly becomes challenging. Nevertheless, these models may prove valuable when it comes to making behaviour predictions because of their expressiveness and will be included in automated processing in future work.

5 Future work

In its current state this work obviously has limitations that need to be addressed in the future:

Given the eye tracking data, sequences are generated by regarding changes in fixated AOI as an observation. While this approach has the advantage of a significant complexity reduction and leads to a manageable observation vocabulary, the opportunity to take the fixation duration into account is missed. Arguably fixation durations can be used to weight the corresponding observation. One possible application of this weighing would be to specify a threshold which has to be exceeded in order for an observation to be considered as a part of an input sequence. Semantically this would mean that very short glances at areas of the application would not be considered to be an input while building the model. It is believed that this approach could reduce noise in the data, therefore, improving overall models' quality.

To further the idea of cleaning noisy eye tracking logs, system logs will also be used in the future. The main idea is that only observations present both in eye tracking and system logs will be considered. For example, if a sequence of *Query* fixations is observed but the corresponding time frame in the system log lacks a 'query submitted' event the observation might be considered as noise. On the other hand, the user might also be engaged in the process of relevance judgement using the query terms as a reference. Alternatively, the user could also be picking up potential query terms from the result documents as was shown by Eickhoff et al. [6] leading to a repeated change of focus between *Query* and *Result* AOI. Both interpretations would make the query observation a valid one for the purpose of model generation. It remains to be closely examined whether models created by combining logs really outperform the simpler versions.

From interpreting the start state probability distribution the hypothesis emerged that especially the beginning of sequences is to be considered as noise. To avoid this problem future model generation could begin when a sequence of consecutive query observations is detected, assuming the user has then finished orientation and task execution has begun. When system logs are included in the analysis the start point could be defined as the time of first query submission.

Regardless of the exact criteria of model generation, the key research question remaining to be examined in future work is how user guidance practically can be achieved. A first approach is likely to follow an idea by Kriewel [11], who implemented strategic system support using a case-based reasoning approach. Using pre-generated HMM a system could detect a user's current situation and give support accordingly, e.g. if based on calculated task completion time it is believed to be more efficient to formulate a new query instead of further examining the result list. Of course this raises the issue of how exactly to include times in the model generation process, which is still an open issue.

Finally, after solving these problems and setting up a set of models using appropriate data sets, a side by side comparative evaluation can be conducted. It is believed that users benefiting from system support will outperform their comparison group in terms of search efficiency and success.

References

1. Ageev, M., Guo, Q., Lagun, D., Agichtein, E.: Find it if you can: a game for modeling different types of web search success using interaction data. In: Proc. 34th ACM SIGIR. pp. 345–354 (2011)
2. Azzopardi, L.: The economics in interactive information retrieval. In: Proc. 34th ACM SIGIR. pp. 15–24 (2011)
3. Bates, M.J.: The design of browsing and berrypicking techniques for the online search interface (1989), https://pages.gseis.ucla.edu/faculty/bates/berrypicking.html
4. Bates, M.J., Bates, M.J.: Where should the person stop and the information search interface start? In: Information Processing and Management. pp. 575–591 (1990)
5. Chau, C.W., Kwong, S., Diu, C.K., Fahrner, W.R.: Optimization of hmm by a genetic algorithm. In: Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on. vol. 3, pp. 1727–1730 vol.3 (Apr 1997)
6. Eickhoff, C., Dungs, S., Tran, V.T.: An eye-tracking study of query reformulation. In: In Proceedings of the 38th Annual International ACM Conference on Research and Development in Information Retrieval. SIGIR '15 (2015)
7. Fuhr, N.: A probability ranking principle for interactive information retrieval. Information Retrieval 11(3), 251–265 (2008)
8. Hassan, A., Jones, R., Klinkner, K.L.: Beyond dcg: user behavior as a predictor of a successful search. In: Proc. 3rd ACM WSDM. pp. 221–230 (2010)
9. He, Y., Wang, K.: Inferring search behaviors using partially observable markov model with duration (pomd). In: Proc. 4th ACM WSDM. pp. 415–424 (2011)
10. Jarrow, R., Lando, D., Turnbull, S.: A markov model for the term structure of credit risk spreads. Review of Financial studies 10(2), 481–523 (1997)
11. Kriewel, S.: Unterstützung beim Finden und Durchführen von Suchstrategien in Digitalen Bibliotheken. Ph.D. thesis, University of Duisburg-Essen (2010)
12. Krogh, A., Larsson, B., Von Heijne, G., Sonnhammer, E.: Predicting transmembrane protein topology with a hidden markov model: application to complete genomes. J. molecular biology 305(3), 567–580 (2001)
13. Liu, Y., Gao, B., Liu, T.Y., Zhang, Y., Ma, Z., He, S., Li, H.: Browserank: letting web users vote for page importance. In: Proc. 31st SIGIR. pp. 451–458. ACM (2008)
14. Luo, J., Zhang, S., Dong, X., Yang, H.: Advances in Information Retrieval: 37th European Conference on IR Research, ECIR 2015, March 29 - April 2, 2015. Proceedings, chap. Designing States, Actions, and Rewards for Using POMDP in Session Search, pp. 526–537. Springer Int. Publishing, Cham (2015)
15. Sonnhammer, E., Von Heijne, G., Krogh, A., et al.: A hidden markov model for predicting transmembrane helices in protein sequences. In: Ismb. pp. 175–182 (1998)
16. Tran, V.T., Fuhr, N.: Using eye-tracking with dynamic areas of interest for analyzing interactive information retrieval. In: Proc. 35th ACM SIGIR. pp. 1165–1166 (2012)
17. Tran, V.T., Fuhr, N.: Markov modeling for user interaction in retrieval. In: SIGIR 2013 Workshop on Modeling User Behavior for Information Retrieval Evaluation (MUBE 2013) (August 2013)
18. Turner, C., Startz, R., Nelson, C.: A markov model of heteroskedasticity, risk, and learning in the stock market. J. Financial Economics 25(1), 3–22 (1989)
19. Wang, K., Gloy, N., Li, X.: Inferring search behaviors using partially observable markov (pom) model. In: Proc. 3rd ACM WSDM. pp. 211–220 (2010)
20. Yue, Z., Han, S., He, D.: Modeling search processes using hidden states in collaborative exploratory web search. In: Proc. 17th CSCW. pp. 820–830 (2014)