Inferring Search Behaviors Using Partially Observable Markov (POM) Model

Kuansan Wang Nikolas Gloy Xiaolong Li
ISRC, Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA

ABSTRACT

This article describes an application of the partially observable Markov (POM) model to the analysis of a large scale commercial web search log. Mathematically, POM is a variant of the hidden Markov model in which all the hidden state transitions do not necessarily emit observable events. This property of POM is used to model, as the hidden process, a common search behavior that users would read and skip search results, leaving no observable user actions to record in the search logs. The Markov nature of the model further lends support to cope with the facts that a single observed sequence can be probabilistically associated with many hidden sequences that have variable lengths, and the search results can be read in various temporal orders that are not necessarily reflected in the observed sequence of user actions. To tackle the implementation challenges accompanying the flexibility and analytic powers of POM, we introduce segmental Viterbi algorithm based on segmental decoding and Viterbi training to train the POM model parameters and apply them to uncover hidden processes from the search logs. To validate the model, the latent variables modeling the browsing patterns on the search result page are compared with the experimental data of the eye tracking studies. The close agreements suggest that the search logs do contain rich information of user behaviors in browsing the search result page even though they are not directly observable, and that using POM to understand these sophisticated search behaviors is a promising approach.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentations.

Keywords

Eye tracking, Partially Observable Markov model, search log mining, segmental Viterbi algorithm, web search behaviors

1. INTRODUCTION

Embedded in the massive log data that capture the detailed interactions between the Web search engine and its users are the insights and knowledge that hold the key to further understand and improve almost every aspect of the search engine. Mining the search logs has been a prevalent and fruitful practice for many applications, ranging from improving retrieval quality [1][5][21][30], query suggestion and clustering [6][34], to inferring contextual user behaviors and document importance [2][16][26][31][36]. Web search logs, however, can only record observable user actions such as clicks and query refinements. How the search result are read and perceived by the users remain elusive. In interpreting the logs, one inevitably has to make certain assumptions on the unrecorded and unobservable aspects of the user behaviors. The impacts of these unobserved behavioral assumptions on the validity of the conclusions can be significant and may well vary from applications to applications.

Eye tracking has been a widely accepted method to study user behaviors that would be otherwise difficult to observe. Many have included this technique to further analyze user behaviors for web search. For example, it has been shown that users read the search results in an uneven fashion, with results on top of the page receiving more attention than the others [18][22][25][27][28]. When the target results are manipulated to appear at a less prominent position, they are rarely seen and, as a result, the user's search success rates deteriorate significantly [18]. Other factors identified by the eye tracking experiments that greatly affect user behaviors on the search engine result page (SERP) but are challenging to directly record or deduce from the search log include the gender differences [25][27], the query intents [33], the habitual preferred scan paths [28], the effects of the search engine brand [27], and the contextual snippets on informational and navigational queries [14].

The knowledge discovered from the eye tracking studies has also inspired numerous log mining algorithms, information retrieval theories and system designs. For instance, the search browsing models that explicitly distinguish the document relevance from the snippet relevance in analyzing the clickthrough data [16] can find physical supports from the experimental results reported in [14][27][28]. The use of the reading time as a proxy to assess the document relevance in [1][17][24][34] is consistent with the observations reported in [33], and the observations in [14] support the "trust bias" theory proposed in [22]. Perhaps the most profound implication from the eye tracking experiments is a plausible explanation to the positional bias that is widely known to exist in the clickthrough data. Positional bias refers to the phenomenon where the search results displayed more prominently on a SERP will receive a higher clickthrough rate, even when the search results are manipulated to display in the reversed order [22][27]. This bias has been a challenge in applying the clickthrough data to assess document relevance because it suggests there are factors affecting the click behaviors other than the relevance of the search results alone. One leading explanation is the uneven browsing pattern of the SERP discovered through eye tracking studies. As

WSDM'10, February 4–6, 2010, New York City, New York, USA.

users are not likely to click on the results they have not even seen, it is quite reasonable to postulate that the user's uneven browsing pattern on the SERP directly impacts the click behavior and becomes a key factor underlying the positional bias. For applications that use the clickthrough data as a form of relevance feedback, adjusting for the positional bias is a critical step in order to correctly infer what the feedback is. A widely used approach is the "depth-first" behavior model inspired by the eye tracking studies [22][25] that are first adopted for web search in [29] and then search advertisements in [32]. Essentially, the depth-first model assumes that users scan the SERPs from top to bottom sequentially, with the relevant search results being clicked as soon as they are read by the users. A click at the position k, for instance, implies the search results preceding that position have all been viewed. It however does not imply the result at $k+1$ is viewed if no click is observed. By considering the clickthrough rate as a form of user feedback only for results that are viewed, this simple model has been shown as effective in filtering out the bias and gathering accurate evidence for improving ranking for search [1][2][29] or advertisements [10][32], detecting adversarial traffic [9], and estimating the user perceived retrieval quality [11][13][26][34].

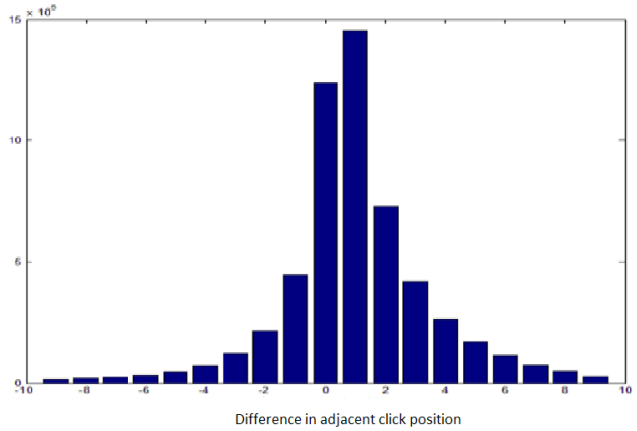

Figure 1: A histogram of the position differences between two adjacent clicks in the search log.

Despite these successes, the depth-first model is a simplification that deserves a closer examination. First, the estimates of the user population following the depth-first strategy vary significantly in the eye tracking literature. While the experimental results of [25] report the percentage can be as high as 65%, others put it as low as 20% [27][28]. Variability in human subjects aside, the SERP browsing behavior is difficult to describe in a precise quantitative manner. This is because analyzing the scanpaths in eye tracking studies remains a formidable challenge as the supports in capturing the fine grained temporal information remain less than ideal [27]. As shown in Figure 1, a typical search log consists of more than 30% of clicks that occur at positions higher or equal to the click immediately preceding, suggesting a significant number of clicks occur after users having clicked search results further down in the SERP, contradicting the assumption of the depth-first behavior model. Secondly, most eye tracking experiments were conducted in a setting where the SERPs presented to the user are highly controlled that frequently occurring elements such as advertisements and query suggestions are all carefully filtered out. It is therefore unclear how the lessons learned in these lab experiments can be extrapolated to the general usages. Third, the experiments were typically conducted on sizeable yet demographically skewed user groups where the subjects were predominantly young college students well versed in technologies. It is reasonable to question whether some of the observed behaviors are representative for the general population. As an example, while these studies typically found viewing more than one SERP as a very rare event, data from our web search logs and from many commercial monitoring services suggest that viewing multiple SERPs still accounts for a significant portion of usages. Therefore, it is highly desirable to have a means to study the subject in the web scale that can go beyond the oft highly controlled lab settings in order to obtain a holistic view of the user behaviors on the SERPs.

This paper describes a mathematical framework, called the partially observable Markov (POM) model, which we use as a complementary method to the eye tracking experiments to uncover unobservable search behaviors. POM tackles the key challenge of using the search logs, i.e., coping with unknown biases and unobservable behaviors, by treating them as a statistical hidden data problem [15]. We then solve for the maximum likelihood (ML) solution of the hidden data, i.e., obtaining the model that can explain the log data with the highest probability. This data mining approach enables us to study how users interact with a search engine at a larger scale and for wider demographics without the logistic constraints of conducting eye tracking experiments. A contribution of this work is, by modeling the sequence of user events in Markov chain, we extend the analysis to consider not only the spatial but also the temporal information in the search logs. There are two aspects of the temporal information considered in this work. First, we consider all the user actions recorded in the search logs in a session, including hovering events, page loading and unloading, query reformulation, etc., in addition to the conventional click events. The model is therefore richer than just click behavior analysis and, in a sense, takes advantage of the additional information that can be extracted from a search session in the logs. Secondly, by lining up all the recordable events along the time line to form a holistic view of the user event sequence, the model is able to exploit the information embedded in the temporal order of the events. A natural outcome of this exploitation is the ML estimation of the user scanpaths on the SERP which, as mentioned in [27], remains a key challenge in obtaining from the eye tracking data. Through the use of probabilistic modeling, the variability in user behaviors is naturally taken into account in the probabilistic distribution in a mathematically tractable manner, and qualitative statements on user behaviors are also quantified in the process.

The rest of the paper is organized as follows. In Section 2, we describe POM in detail. We show that a POM model can be regarded as a variant of the hidden Markov model (HMM) where some state transitions do not correspond to observable events. The topic was briefly considered in the seminal work of [3] in which a 'null' transition was introduced into HMM. To facilitate the model training using the well known forward-backward algorithm, certain restrictions on the null transitions (e.g., null transitions cannot be self-looped) were introduced. These restrictions are relaxed in this work, leading the implementation of the forward-backward algorithm rather complicated and infeasible. To address this challenge, we develop a new training algorithm, called the segmental Viterbi algorithm, which is based on segmental decoding and Viterbi approximation. Essentially, the segmental Viterbi algorithm combines two common alternatives to the forward-backward algorithm into one unified framework for ML parameter

estimations. We derive segmental Viterbi in details in Section 2, and show in Section 3 the results of applying the POM model to the search log data collected by a commercial search engine. We compare the estimated ML hidden process with the physical observations reported by the eye tracking studies. The consistencies between the POM and eye tracking results suggest the search logs may well retain enough subtle cues of the user behaviors that can be uncovered by data mining methods. Finally in Section 4 we discuss how the methods described in this paper can be further extended to more applications.

2. PARTIALLY OBSERVABLE MARKOV (POM) MODEL OF SEARCH LOGS

The search logs record detailed interactions between the search engine and the users. Among the information available in the log are: (1) the events triggered by user actions in a search session, such as clicking or hovering over a search result and query submission or reformulations, (2) the actions taken by the search engine in response to the user events, such as the search results shown to the users and their respective locations on the SERP, and (3) a unique identifier associating the users with their search sessions. In short, the search logs tell us *when* and *who* does *what* at *where* on the SERP. Our modeling effort aims at uncovering *how* the sequence of events can be triggered and hopefully provides a glimpse into *why*.

2.1 Partially observable process

The key concept of this model is to treat the user events as a *partially observable* stochastic process. The notion is further illustrated in Figure 2 where the top row represents a search session as recorded in a search log. There, each observable user action, such as a click on a result or a query reformulation, is captured as an event e_i. Between two adjacent events in the observable event sequence $O = \{e_1, e_2 \ldots\}$, the user may have viewed and skipped many search results that cannot be recorded in the logs. As a result, there are many alternatives that can lead to the same observation, two of which possible hypotheses Q_1 and Q_2 are illustrated in Figure 2 where the unobservable user actions are shown as dotted arrows. More specifically, let V_k be the random process denoting the k-th object (in the temporal order) the user views in a search session. The possible object types include not only the search results but also the query suggestion, spelling correction, advertisements, query refinement input box, and other links in the header and footer of the SERP. For each k, we further define a Boolean random variable S_k representing whether V_k is skipped or not, as shown in Figure 2. The model is partially observable because, in the search logs, we can only record the V_k for which object k is entered text or clicked (i.e., $S_k = 0$), or is hovered but not clicked (i.e., $S_k = 1$). Objects that are viewed and skipped without hovering are not recorded. As such, a given session observation can be a result of theoretically infinite many hypotheses $Q_j = \{(V_1, S_1), (V_2, S_2) \ldots (Vn_j, Sn_j)\}$, each of which has a different session length n_j. In Figure 2, for example, the lengths for the two hypotheses Q_1 and Q_2 are $n_1 = 6$ and $n_2 = 7$, respectively. More generally, an observed user event sequence O can be the outcome of any partial observable sequence that meets the following conditions:

$$V_i = e_m, V_j = e_{m+1}, S_i = S_j = 0 \Rightarrow S_l = 1, \forall m, l, i < l < j$$

We use the notation $Q_k \to O$ for Q_k that can be a partially observable sequence of the user event observation O. Consequently, we have

$$P(O) = \sum_{Q_k \to O} P(Q_k) \tag{1}$$

Here, we regard the "view sequence" $\{V_1, V_2 \ldots\}$ as modeling user's browsing behavior that is analogous to the "scanpath" in the eye tracking studies. On the other hand, the "engagement sequence" $\{S_1, S_2 \ldots\}$ is viewed as modeling the user's skipping or clicking behavior that, conceptually, is intimately related to the user's relevance judgment of the search results. However, since we include the query refinement and search result hovering in our analysis, the engagement variable S_k really models whether a user has interacted with an object on the SERP or not. To fully understand the interactions between the user and the search engine, one can examine for each session all the possible hypotheses, each of which is weighted by its likelihood

$$\begin{aligned} P(O) &= \sum_{Q_k \to O} P(Q_k) \\ &= \sum_{Q_k \to O} P(V_1, S_1) P(V_2, S_2 \mid V_1, S_1) \ldots P(V_{n_k}, S_{n_k} \mid V_1, S_1, \ldots V_{n_k-1}, S_{n_k-1}) \end{aligned} \tag{2}$$

2.2 Model Assumptions

A key challenge in evaluating each search session with *all* of its possible hypotheses is the lengths of these hypotheses are not only unobservable but also unequal. To tackle this problem, we first assume that underneath the partially observable process is a Markov chain of order N, with an example of $N = 1$ turning (2) into

$$\begin{aligned} P(O) &= \sum_{Q_k \to O} P(Q_k) \\ &= \sum_{Q_k \to O} P(S_1 \mid V_1) P(V_1) \prod_{j=2}^{n_k} P(V_j \mid V_{j-1}, S_{j-1}) P(S_j \mid V_j, V_{j-1}, S_{j-1}) \end{aligned} \tag{3}$$

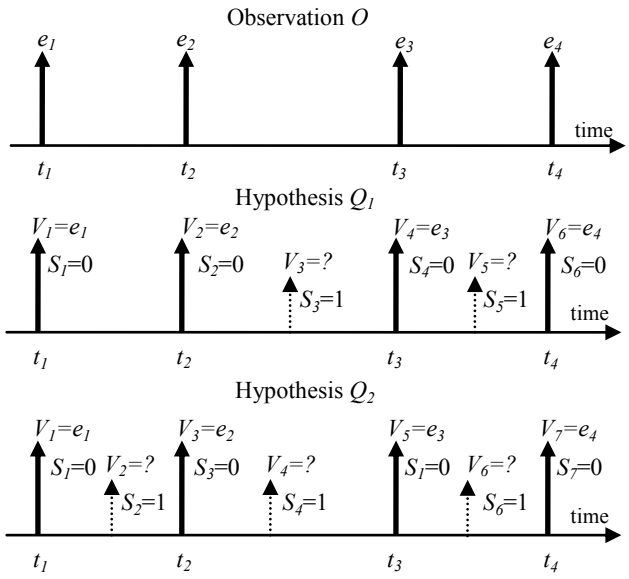

Figure 2: Illustration of two partially observed hypotheses that can both give rise to the observed sequence at the top row.

Although it is possible to directly work with a Markov chain of any order (Sec. 4), in this work we consider only first order Markov model for simplicity.

Similarly, we further make the following two assumptions not for theoretical necessity but for engineering simplicity. First, we assume that the engagement variable S_k is only dependent upon the SERP object the user is currently viewing, i.e., S_k has V_k as the sufficient statistics such that

$$P(S_j \mid V_j, V_{j-1}, S_{j-1}) = P(S_j \mid V_j) \tag{4}$$

Secondly, we assume that the user's mental process in determining which SERP object to visit next is not impacted by whether the user has engaged with the current search result or not, i.e.,

$$P(V_j \mid V_{j-1}, S_{j-1}) = P(V_j \mid V_{j-1}) \tag{5}$$

This assumption appears inconsistent with the cascade model [13] and its derivatives [11][19] in which the human users are believed to be more likely to continue exploring the rest of the SERP when the search results they encounter are not relevant. However, it has been suggested [34] that the relevance of the result is a very weak predictor of user's continuing exploration on the SERP and vice versa. While the intuition behind the cascade model holds for some cases, many search sessions see the opposite user behavior. Most notably, instead of reading the next result, many quickly abandon the SERP by reformulating the queries when they feel the search engine does not fully understand their search intents. Conversely, many users continue reading the rest of the SERP not because they have not encountered relevant results but because they see the search engine has retrieved useful leads, especially for informational queries that can be best served by multiple results. Since the behavioral data suggest what users read next can result from diametrical causes, we in this work follow the modeling approach in [34] and assume user's continuing browsing pattern is statistically independent of the result being relevant or not.

With these assumptions, we can further simplify (3), the likelihood of a partially observed hypothesis as

$$\begin{aligned} P(O) &= \sum_{Q_k \to O} P(Q_k) \\ &= \sum_{Q_k \to O} \prod_{j=1}^{n_k} P(V_j \mid V_{j-1}) P(S_j \mid V_j) \end{aligned} \tag{6}$$

where we use V_0 to represent the submitted query received at the onset of a search session. From (6), it is clear that the statistical properties of a POM model are fully specified by the view transition probabilities $v_{mn} = P(V_j = n \mid V_{j-1} = m)$ and the engagement probabilities $s_m = P(S_j \neq 0 \mid V_j = m)$. In the following, we use Λ to denote the collection of all these transition and engagement probabilities that parameterize a POM model.

2.3 Segmental Decoding

A key utility behind POM is to allow statistical inference on how users browse the SERP. We following the convention of HMM and use the term "decoding" to refer to the process of uncovering the maximum likelihood (ML) hypothesis Q_k from an observed sequence O, namely, given a POM with parameter Λ the decoding process is to find

$$\hat{Q}_k = \arg\max_{Q_k \to O} P(Q_k \mid \Lambda) \tag{7}$$

The decoding problem highlights a major difference between a POM and a HMM in that the decoded sequence in a POM has an unknown length, namely, the n_k in (6) is itself a hidden variable in a POM whereas it is a part of the observation in a HMM. To address this issue, we propose the segmental decoding technique, adapted from the segmental model of speech recognition [20], for decoding a POM model. The segmental decoding technique is governed by three basic principles that effectively tackle the problem of having to deal with the theoretically unlimited number of hypotheses due to the unknown length n_k in (6). First, the *concatenation principle* states that the ML solution to (7), with the first order Markov assumption, is simply a concatenation of the ML solutions between two adjacent observations e_j and e_{j+1}. More precisely, if we denote

$$\hat{Q}_{k_j}^{(j)} = \arg\max_{Q_k \to (e_j, e_{j+1})} P(Q_k \mid \Lambda)$$

We have $\hat{Q}_k = (\hat{Q}_{k_0}^{(0)}, \hat{Q}_{k_1}^{(1)}, \ldots)$. In this work we introduce an "end-of-session" object and pad it to the end of every observation so that (7) can be realized using segmental decoding technique even for the last segment. Secondly, the *recursive principle* of segmental decoding notes that the concatenation principle can be recursively applied to itself. Suppose we have a hypothesis in which the user is assumed to have viewed and skipped a result x between e_j and e_{j+1}. We can apply the same arguments underlying the concatenation principle and obtain

$$\hat{Q}_{k_j}^{(j)} = \left(\arg\max_{Q_k \to (e_j, x)} P(Q_k \mid \Lambda), \arg\max_{Q_k \to (x, e_{j+1})} P(Q_k \mid \Lambda) \right)$$

Finally, the *parsimonious principle* states that the ML solution cannot afford loops as hidden sequence, i.e., we can discard any Q_k that hypothesizes $x = e_j$. This is because (6) shows that incorporating the probability of any repeated path can only lower the overall likelihood score. The sequence length of any $\hat{Q}_{k_j}^{(j)}$ is therefore bounded by the maximum number of the result objects that can appear on the SERP. Accordingly, the segmental decoding can be realized with straightforward dynamic programming with a finite search space.

In practice, segmental decoding is not necessarily limited to obtaining only the maximum likelihood hypothesis. Many dynamic programming algorithms allow decoding of the top N hypothesis for each observation, an enhancement commonly called as the N-best decoding [20]. In this work, we consider only N = 1 for analyzing the search log, but use N = 5 in the Viterbi training process described below.

2.4 Viterbi Training of a POM model

The search logs consist of a set of observations $\{O_1, O_2 \ldots\}$ that are collected independently. The task of training a POM is to find an estimation of the model parameter that is optimal in the ML sense, namely,

$$\hat{\Lambda} = \arg\max_{\Lambda} \prod_i P(O_i \mid \Lambda) = \arg\max_{\Lambda} \prod_i \sum_{Q_k \to O_i} P(Q_k \mid \Lambda) \tag{8}$$

As POM being in the class of the hidden data problem, the training intuitively can be achieved by many well known algorithms such as forward-backward [3] or the Expectation-Maximization (EM) algorithm [15]. The key idea behind either of these algorithms is to start an initial guess of the parameter $\Lambda^{(0)}$ and gradually improve the ML estimation through iterations. More specifically, in the l-th iteration, the EM algorithm aims to gradually improve the model by finding the re-estimation formula through

$$\hat{\Lambda}^{(l)} = \arg\max_{\Lambda} \sum_i P(O_i \mid \Lambda^{(l-1)}) \log P(O_i \mid \Lambda)$$

Since the summation terms are bounded by Gibb's inequality, we note that the maximization problem can be solved by using the condition in which the equality holds for Gibb's inequality. The re-estimation formula for (8) therefore amounts to

$$\hat{v}_{mn}^{(l)} = \frac{\sum_i \sum_{Q_k \to O_i} P(Q_k \mid \Lambda^{(l-1)}) \sum_{j=1}^{n_k} 1(V_j = m, V_{j+1} = n)}{\sum_i \sum_{Q_k \to O_i} P(Q_k \mid \Lambda^{(l-1)}) \sum_{j=1}^{n_k} 1(V_j = m)}$$

$$\hat{s}_m^{(l)} = \frac{\sum_i \sum_{Q_k \to O_i} P(Q_k \mid \Lambda^{(l-1)}) \sum_{j=1}^{n_k} 1(V_j = m, S_j \neq 0)}{\sum_i \sum_{Q_k \to O_i} P(Q_k \mid \Lambda^{(l-1)}) \sum_{j=1}^{n_k} 1(V_j = m)} \tag{9}$$

Here we use 1(.) to denote the indicator function that is defined as:

$$1(x) = \begin{cases} 1 & x \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

As is in the case of decoding, the unobservable nature of session length n_k makes it challenging to apply exactly the EM algorithm to train a POM model as there will be infinite number of Q_k in (9) to be considered. To address this problem, we use the expedited approximation known as the Viterbi algorithm [20] that only uses the top N-best hypotheses in the training. As a result, the Viterbi training process for POM consists of the following steps:

1. **Initialization**: Start with an initial model $\Lambda^{(0)}$ (Sec. 2.4.1).
2. **Decoding**: For iteration l, apply the segmental decoding method (Sec. 2.3) with $\Lambda^{(l-1)}$ to find top N hypotheses for each observation.
3. **Re-estimation**: Obtain $\Lambda^{(l)}$ using (9).
4. **Iteration**: repeat step 2 and 3 until the process converges.

Theoretically, N-best Viterbi training will asymptotically approximate the EM algorithm as N becomes very large. With a finite N, however, Viterbi training does no longer enjoy the mathematically guaranteed convergence property of the EM algorithm. In our applications, we have found that with N set to 5, the training process does converge (with proper initial conditions, Sec. 2.4.1) in less than 8 iterations for sizeable log data, consistent with the empirical findings in applying the Viterbi training to HMM training in speech recognition.

2.4.1 Model Initialization

Both EM and Viterbi are iterative algorithms that converge only to a local optimum. It has been empirically observed that the initial condition of these iterative algorithms plays an important role in the quality of the model the algorithms eventually converge to. How to best choose an initial condition, however, remains an open research question. In this work, we experimented with two approaches to initialize the POM model. The first approach initializes the view sequence for each observation using the depth-first model (Sec.1) with probability 1. For example, if we observe a search session consists of a click on the third search result, we hypothesize the user has also viewed the first and the second results and decided to skip both of them. The initial POM model $\Lambda^{(0)}$ is then obtained by counting the ML view transition and skip probabilities with (9).

The second approach detaches itself farther from the depth-first model and solve for the initial condition by considering the transition probability of adjacent events $c_{ij} = P(e_{t+1} = j \mid e_t = i)$ that can be counted directly from the search logs. We note the hypothesis that the user does not view and skip anything in between has the probability $v_{ij}(1 - s_j)$, and the probability of the user viewing and skipping one intermediate result k before reaching result j (and not skipping it) is $\sum_k v_{ik} s_k v_{kj} (1 - s_j)$. With induction, the probability of c_{ij} is a sum of the conditional probabilities of users skipping exactly 0, 1, 2… results between interacting with result i and j. Assuming each condition weighs equally, we have

$$c_{ij} = v_{ij}(1 - s_j) + \sum_k v_{ik} s_k v_{kj} (1 - s_j) + \sum_m \sum_n v_{im} s_m v_{mn} s_n v_{nj} (1 - s_j) + \ldots \tag{10}$$

To simplify the notation, we can rewrite (10) in the matrix form

$$\begin{aligned} \mathbf{C} &= \mathbf{V}(\mathbf{I} - \mathbf{S}) + \mathbf{VSV}(\mathbf{I} - \mathbf{S}) + (\mathbf{VS})^2 \mathbf{V}(\mathbf{I} - \mathbf{S}) + \ldots \\ &= (\mathbf{I} + \mathbf{VS} + (\mathbf{VS})^2 + \ldots)\mathbf{V}(\mathbf{I} - \mathbf{S}) \\ &= (\mathbf{I} - \mathbf{VS})^{-1} \mathbf{V}(\mathbf{I} - \mathbf{S}) \end{aligned}$$

or equivalently,

$$\mathbf{C}(\mathbf{I} - \mathbf{VS}) - \mathbf{V}(\mathbf{I} - \mathbf{S}) = \mathbf{0} \tag{11}$$

where $\mathbf{C} = [c_{ij}]$, $\mathbf{V} = [v_{ij}]$, and $\mathbf{S} = \text{diag}(s_i)$, respectively. This equation highlights the fact that there exists a boundary condition for which the parameters must satisfy, i.e., although POM has two hidden matrices **V** and **S** to consider, we only have a *single* degree of freedom in choosing their initial values. In our implementation, we typically choose a diagonal matrix $\mathbf{S} = \alpha \cdot \mathbf{I}$ as the initial value and solve for **V** using (11). Because **V** represents a probabilistic transition matrix, each entry in **V** must be non-negative and each row must sum up to 1. Obtaining **V** from (11) is therefore a constrained problem solvable using the gradient descent algorithm.

Empirically, we have found that the second approach, with α set at the pSkip value of the whole search log [34], yields a much better initial condition than the depth-first model in terms of the number

of iterations needed to converge and the smoothness of the probabilities in the converged results.

3. COMPARISONS WITH EYE TRACKING RESULTS

The research question explored in this paper is to what extent the mathematical framework of POM can uncover the user activities that are not recorded in the search logs. Since traditionally such activities are studied through direct observations such as experiments using the eye-tracking devices, we compare the results obtained by the POM model with those in the published eye tracking findings.

3.1 Data Collection and Model Training

We apply the POM model to the search log data collected by a commercial web search service deployed in the EN-US market that used a two column SERP layout as shown in Figure 3. The core search results (CR) were displayed on the left column, where the right column displayed the query suggestions (QS). When appropriate, advertisements could occupy the top portion of the left column (labeled as TA) or the lower portion of the right column (labeled as SA), right below the query suggestions, as shown in Figure 3.

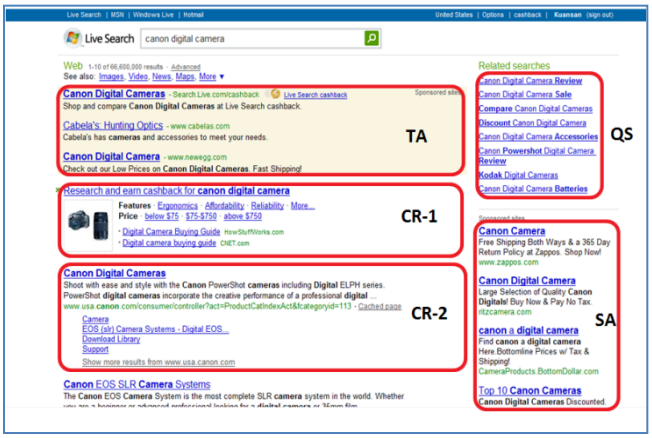

Figure 3: A two column SERP layout showing content modules top ads, core results, query suggestions, side ads labeled as TA, CRs, QS, and SA, respectively. The second text box for query reformulation and the paging buttons located near the bottom of the SERP are shown here.

All these content modules can contain more than one search result in them. For example, Figure 3 shows the TA module has three results. In addition, two textbox objects, one just below the header and the other just above the footer, and a few paging buttons were also available on the SERP for the users to reformulate their queries or navigate to the next SERP. The usages of these textboxes and paging buttons are also captured in the search logs and considered in the POM analysis. Because not all the content modules would appear on every SERP, the search logs also recorded their presence. We utilized this information during the decoding process (Sec. 2.3) so that the probabilities of the infrequently appearing modules would not be under-estimated. Similarly, as clicking on navigational objects such as QS items or the paging buttons will necessarily take the users to a different SERP, special cares were taken to keep track of the navigational activities so that any consequent click on the back button to return to the original SERP could be correctly recorded in the search log as resumed viewing activities. We implemented the decoding and training algorithms described in Sec. 2 for search logs of various durations, ranging from one full day to two months. Typically, the segmental Viterbi training takes five to eight iterations to converge on a week long search log. We found the duration of the training logs do not change the outcomes dramatically. The results reported below are based on the parameters estimated with one week worth of the search data.

3.2 Comparison to Scanpath Analysis

A key question that the eye tracking experiments aims to study is the manner users pay attention to the results on the SERP. This issue has been studied by analyzing the scanpath that traces the order of the search results receiving gaze fixations. We compare the scanpath in eye-tracking studies to the ML hypothesis $\hat{Q}_k$ decoded from the search logs using (7). The predominant model of the SERP scanpath is the depth-first model based on the eye-tracking experiments reported in [25]. It states that search users read the results sequentially in the order as presented, clicking on relevant results as soon as they are encountered, rather than survey the whole SERP before making any selections.

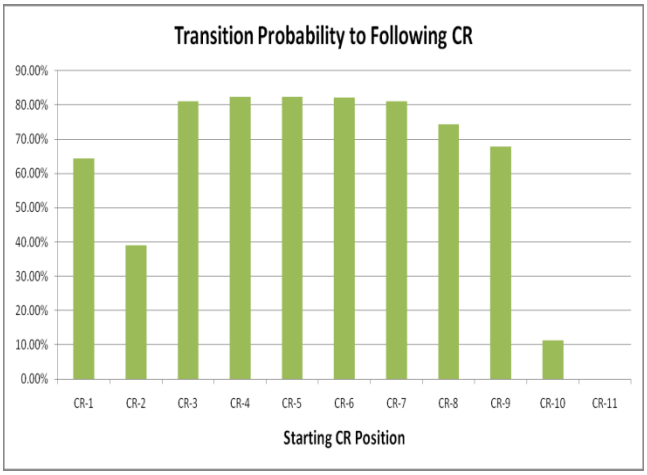

Figure 4: View transition probabilities of CR-n to CR-(n+1) in POM

The studies behind the depth-first model, however, did not include the cases where advertisements may appear on top of the core search results. A direct application of the depth-first model to SERPs where the TA module exists would suggest that top ads be the first module the users see. After training from the logs, however, POM shows the probabilities of the first module seen by the users are 0.92 for CR-1 and only 0.036 for the TA. In other words, POM suggests the most likely user behavior is the top ads are largely ignored. The result is consistent with the "banner blindness" effects that are observed in the eye tracking studies for general web page advertisements [7]. Similarly, the probability of users viewing the QS module first, positioned on the top on the right column, is 0.030. POM analysis therefore suggests that, upon the page load, users most likely go directly to the core search results and ignore other areas on the SERP even though their locations are no less prominent.

The view transition probabilities within the CRs, shown in Figure 4, illustrate further comparisons between POM inferred browsing

behaviors and the depth-first model. As shown in the first two columns of Figure 4, the POM transition probabilities from CR-1 to CR-2 and from CR-2 to CR-3, are only 0.64 and 0.39, respectively, suggesting that at the top of the SERP, users are not mechanically following the depth-first model of the reading behavior. A deeper investigation into the data shows a considerable portion of the probability mass here goes to the transition into query reformulation, leading to a conclusion that, for SERPs receiving no clicks, the ML decoded hypothesis consists of only top two core results being viewed before users either abandon or reformulate their queries. This finding matches the eye tracking observations that report users do not explore further results if top two or three results are not relevant for Yahoo and Google, respectively [27].

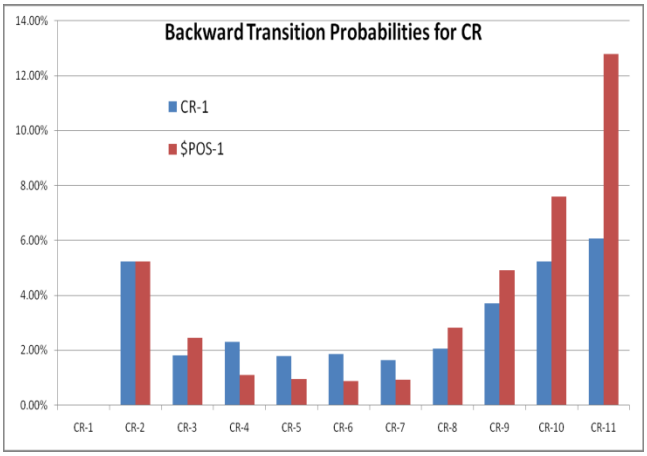

Figure 5: Probabilities of backward viewing the previous result (labeled as $POS-1) and backward viewing the top result (labeled as CR-1) with respect to search result positions in POM

Based on the data collected in the logs, the scroll line of the SERP we studied straddles between CR-2 and CR-4 for the majority of the cases, as Figure 3 demonstrates. Below the scroll line, the POM view transition probabilities suggest the depth-first viewing pattern is indeed the most dominant behavior. Figures 4 shows POM infers from the logs that more than 80% of chances users simply read sequentially from CR-3 down, in a sharp contrast to the view transitions above the scroll line. The dramatic impact of the scroll line on the SERP viewing behaviors are also reported in the eye tracking studies [22]. As a matter of fact, POM seems to have inferred a more detailed and intricate relationships between the viewing behaviors and their SERP location dependency.

First, the series of experiments [22][27][28] report that it is not unusual to see gaze fixations on search results that have already had fixations before, implying that users occasionally look back at results they have previously viewed. ML estimates from the POM model indicate the probabilities of looking backward are position dependent, and the most dominant trend is to either transition back one result or all the way back to the first CR on the SERP. As shown in Figure 5, backward transition probabilities are lowest for results that are just below the scrolling line, and gradually increase towards the end of the SERP where we also observe the backward transition becomes more local, namely, users seem to look one result back rather than jump back to the first CR. We are not aware of any eye tracking studies that report quantitative details on backward viewing behavior to corroborate the ML estimates of the POM model.

The presence of the right column on the SERP does not seem to command a lot of attention. This is shown in Figure 6 where the transition probabilities from core results to the right column are in general small with the exception at CR-2, right above the scroll line, where the probability of viewing query suggestion next peaks at 11.2%. This POM inferred behavior is consistent with the intuition that scrolling the SERP takes more user effort than scanning the right column. Nevertheless, we note that, as shown in Figure 4, the probability of viewing CR-3 right after CR-2 is still three times more likely than sidetracking to the right column, suggesting the sequential depth-first model still describes the dominant behavioral pattern. The transition probabilities from the right to the left column are all very low, with the highest number at 8.24% from SA to CR. The POM ML estimates indicate that once users are in the right column they are most likely to reformulate the query or end the search session.

Aberration from the depth-first viewing, either through query reformulation (labeled as ReQ in Figure 6), paging, or diverting to the right column, has the lowest probability right below the scroll line. As can be seen by combining the readings from Figures 5 and 6, the ML estimates from the POM model suggest that, at the bottom of the SERP, it is very likely for the users to look back one result before traversing down the SERP and clicking on the paging button to visit the next page or entering a reformulated query into the textbox located below the paging section. This inferred behavior is similar to the backward viewing behaviors described above. Comparing the ML estimates from POM model with the simplified depth-first model derived from the eye tracking data, we observe that the two models least agree on the behaviors towards either end but are largely consistent in the middle of the SERP. Because the SERP viewing behavior seems to be position dependent, statements about the percentage of users following the depth-first model cannot be made summarily. This might be an explanation to the large discrepancies reported in the literature (e.g., [25] vs. [27])

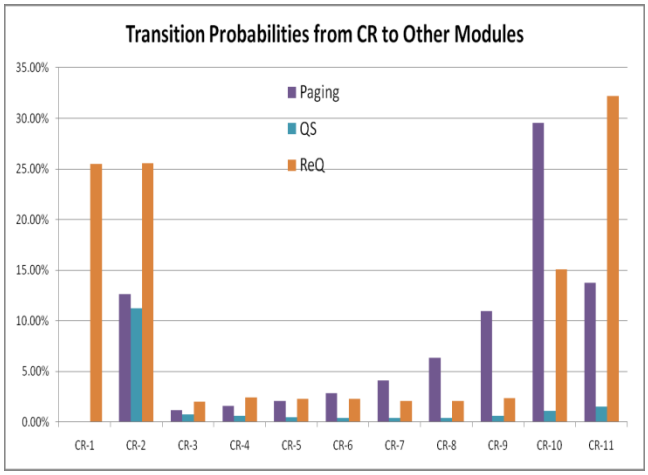

Figure 6: POM inferred probabilities of not continuing to read core results with respect to SERP positions

3.3 Click and View Positional Bias

Eye tracking studies are instrumental in identifying an important web browsing behavior that all the presented results are not equally read by the users [28]. As users are likely to only click on the

results they have viewed, this uneven browsing behavior has tremendous implications on correctly interpreting the clicks in the web logs in general and search logs in specific. The eye tracking results reported by Joachims *et al* [22] demonstrates the positional bias on core search results both in terms of the search results being viewed and clicked. Based on the experiments conducted on users' viewing of Google SERP, their data showed the chance of search results being seen reduces dramatically as their rank positions increase along the length of the SERP. For example, the chance of having a gaze fixation on CR-4 is only half of that on CR-1. It stands to reason that, if for a query CR-1 is merely receiving the same clickthrough rate as CR-4, it is unfair to characterize CR-1 as being equally relevant as CR-4. In fact, CR-4 is likely to be more relevant than CR-1 because it receives more clickthrough per impression. Similarly, Joachim's experimental data seem to show the manner of presentation has significant effects on the clickthrough rate as the clickthrough rate briefly trends up around CR-6 where the screen scroll line lies in those controlled experiments [22].

To compare POM with the fixation rates from the eye tracking data, we show in Figure 7 the aggregated view and clickthrough rates for the CR module derived from the POM model analysis. The aggregated view and clickthrough rates are obtained by running all the search sessions in the logs through the POM ML decoder (Sec. 2.3) and counting the *Boolean* frequency of each CR being viewed and clicked per SERP, respectively. By "Boolean frequency" we mean a search result is counted as a view/click, respectively, if it appears in the decoded sequence, regardless how many times it occurs. As can be seen from Figure 7, the POM inferred view rates bear close resemblances to the eye tracking data reported in [22] as they also show the strong positional bias in terms of the CRs being viewed and clicked based on its position on the SERP.

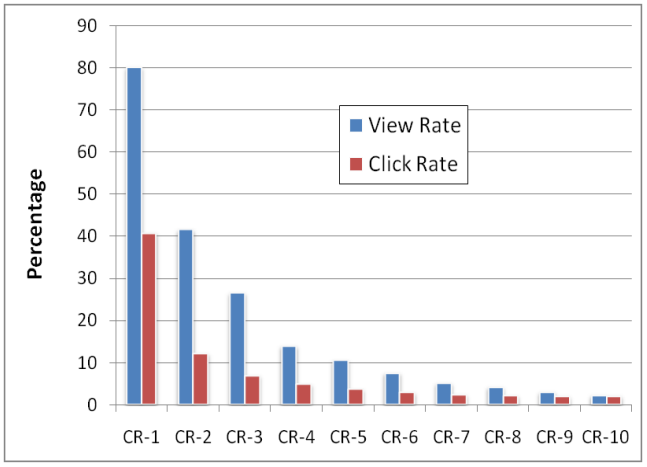

Figure 7: POM inferred view rate and the clickthrough rate for CRs with respect to their positions on the SERP. Like eye tracking experiments, POM inferred behaviors also suggest positional bias in both SERP viewing and clicking behaviors.

This agreement on positional biases is noteworthy especially given the experimental conditions behind the data are significantly different. The POM inferred data are based on the field deployment where noncore search results are not filtered out from the top and on the side of the SERP, in a contrast to the eye tracking studies. Secondly, the search engine used for POM analysis tend to put more detailed descriptions for the search results than Google, the search engine used for the eye tracking studies that is known to have a shorter abstract for each search result so that the screen size could accommodate more search results [27]. Accordingly, the data for POM studies have a variable scroll line position ranging from just below CR-2 to CR-4, depending on whether the TA module is triggered for the queries or not. In contrast, the scroll line used in Joachim's experiments has a fixed position at 6. Perhaps due to these reasons, there is no clear scroll line effect observed in Figure 7. However, both data sets suggest that search results placed under the scroll line have less than 10% of being viewed by the users.

The POM inferred data exhibit subtle differences on the top search results, though. The eye tracking studies report that the first two results are equally viewed, and sometimes users view CR-1 after CR-2 [22][27], which can probably explain the relatively lower fixation rate on CR-1 than in Figure 7. The POM inferred data, however, do not support the same observation in that the view transition from page load to CR-2 directly has very low probability. The POM data suggest the leading causes for CR-1 not being viewed are sessions ending at TA or users clicking on other navigational buttons located in the header section (e.g. image or news search). As a result, Figure 7 shows a much higher CR-1 view rate. The rest of the CR view rates, in the mean time, are considerably lower than the fixation rates in the eye tracking experiments where the data were obtained by carefully filtering out non CRs. In the log data for POM inference, we observe that query reformulation is a very substantial user activity and the high probabilities of query reformulation at the top two positions contribute to the low view rates for the rest of the CRs. Although the eye tracking studies also report that users on the average read only two results before reformulating their queries [27], it is not clear if they occur as frequently in the controlled experiments where the eye tracking devices were used.

Figure 7 also highlights that the view rate is not simply a scale-offset of the click rate, confirming that the click and browsing behaviors are mostly likely not driven by the same cognitive mechanism and that they deserve to be studied separately. The similar observations have also been made by the eye tracking studies in [14][18] where the users are observed to exhibit similar browsing yet quite different clicking behaviors in carrying out informational and navigational queries.

4. DISCUSSION

The consistent conclusions drawn from the statistical POM analysis and the physical eye tracking experiments suggest that data mining on the search logs is a viable approach towards a deeper understanding of the search behaviors, especially in the areas of browsing and clicking behaviors on the SERP. Aside from the qualitative model derived from the eye tracking studies, statistical techniques can provide a quantitative and analytically tractable framework that uses the massive search log data to understand and improve the search engine quality without running into the physical difficulties of use eye tracking devices described in [27].

One promising application is to apply POM to better estimate the search result relevance from the search logs for the purpose of improving search engine ranking function. The key intuition, as widely adopted by [1][2][21][29], is that higher quality search results may lead to a higher clickthrough rate and vice versa. The challenge is how to compute the clickthrough rate correctly. The ability of POM in inferring viewing sequence can be potentially

helpful here, although the topic cannot be fully explored in its entirety in this paper.

Table 1: ML decoded sequence in POM

Observed Click Sequence	POM Decoded Sequence
1, 5, 3	1, 2, **3**, 4, 5, 3
2, 8, 1	**1**, 2, 3, 4, 5, 6, 7, 8 ,**9**, 1

Table 1 illustrates two observed click sequences in the CR module and their corresponding POM decoded view sequences. The first observation involves a backward click on CR-3 after a click on CR-5 is first observed. The common approach, as pioneered in [29], is to treat the search session as having clicks on CR-1, 3, and 5 with CR-2 and 4 skipped. Under this view, the click on CR-3 would carry the exact the same weight as if the observed click sequence were CR-1, 3, and then 5. In contrast, the same observation will elicit a somewhat different interpretation under POM as the decoded sequence in Table 1 shows CR-3 is first viewed, skipped, and only be clicked after the user has viewed CR-5. In other words, POM suggests the user has two impressions on CR-3, and only one out of these two impressions does a click occur. The POM model suggests a lower estimation on the clickthrough rate on CR-3 than CR-1 or 5 in this case. The same additional penalty on the backward click also applies to the second observation in Table 1 in which the last click on CR-1 will be regarded as less significant as the click on CR-2 as the POM model infers the user most likely has skipped CR-1 once before. In addition, the decoded sequence for this observation also highlights the ability of POM to infer the user has most likely viewed an additional search result CR-9 beyond the last clicked position CR-8, enabling us to treat CR-9 as viewed and skipped like CR-3 through 7. It is well known from the eye tracking data [22] that users often read additional results on the SERP beyond the last clicked position, although the pattern is so noisy and position dependent that no simple qualitative description can be made. As demonstrated in Table 1, POM provides a statistical way of modeling this behavior.

Despite these encouraging results, the current formulation of POM can be made more powerful by further relaxing the assumptions described in Sec. 2.2, such as the first order Markov assumption. Recent advancements in machine learning have allowed flexible and potentially infinite order of Markov chain to be used in modeling temporal sequences. Exemplary techniques that can be adapted for POM include the variable N-gram in language modeling [20], variable length HMM [8], and infinite HMM [4] that uses Dirichlet process to integrate out infinite parameters.

The current formulation also uses only one set of probabilities to parameterize the model. As a result, the model is estimating the overall average behaviors for all users and query types. Although some eye tracking studies support that browsing behaviors might not be varying with query types being informational or navigational [14][18][27], others do show subtle behavioral changes between informational and transaction queries [33]. In addition, it is known that male and female subjects have dramatically different search behaviors [27][28]. All these known factors affecting the search behaviors make the current use of a single parameter set less than ideal. A natural extension to (2) is to use a mixture model so that factors leading to significant behavioral differences can be captured individually by mixture components. Again, the number of mixture components does not have to be hardcoded but can be automatically learned using Bayesian techniques such as Chinese Restaurant Process. In addition, the current formulation POM only uses a simple skip probability to model the clicking behavior. It has been argued that the quality of search result snippets and the relevance of the landing page should be further teased apart and modeled separately [16]. However, how to properly model these factors remains an unanswered question and requires more experimentation. As an example, the eye tracking data suggest the criteria to determine whether a snippet is effective seem to be opposite for navigational and informational queries [14]. Understanding the effectiveness of a mixture POM model may prove a good first step in resolving these issues.

5. ACKNOWLEDGMENTS

Bin Cao contributed to the inception and the first implementation of the gradient descent algorithm described in Sec. 2.4.1 during his internship at Microsoft Research Redmond. The authors would like to thank Ed Cutrell, Filip Radlinski, and Bing Search usability team for generously sharing the time, data, and experience in their eye tracking experiments, and Alex Acero, Peter Bailey, Fritz Behr, Nick Craswell, Thore Graepel, Ralph Herbrich, Paul Hsu, Bob Jenkins, Hang Li, Milind Mahajan, Ramez Naam, Bill Ramsey, Bryan Sera, Harry Shum, Toby Walker, Zijian Zheng, and Jingren Zhou for their assistance in search log instrumentation, processing, and technical discussion.

6. REFERENCES

[1] Agichtein, E., Brill, E., and Dumais, S. 2006. Improving web search ranking by incorporating user behavior information. In Proc. SIGIR'2006, New York, NY, 19-26.

[2] Agichtein, E., Brill E., Dumais, S., and Ragno, R. 2006. Learning user interaction models for predicting web search result preferences. In Proc. SIGIR'2006, New York, NY, 3-10.

[3] Bahl, L., Jelinek, F., and Mercer, R. 1983. A maximum likelihood approach to continuous speech recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-5(2), 179-190.

[4] Beal, M. J., Ghahramani, Z., and Rasmussen, C. E. 2001. The infinite hidden Markov model. In Proc. NISP'2001, Vancouver, B. C.

[5] Becker, H., Meek, C. and Chickering, D. 2007. Modeling contextual factors of click rates. In Proc. AAAI'2007, Vancouver, BC, 1310-1315.

[6] Beeferman, D. and Berger, A. 2000. Agglomerative clustering of a search engine query log. In Proc. KDD'2000, New York, NY, 407-416.

[7] Burke, M., Hornof, A., Nilsen, E., and Gorman, N. 2005. High-cost banner blindness: Ads increase perceived workload, hinder visual search, and are forgotten. ACM Trans. on Computer-Human Interaction (TOCHI), 12(4), 423-445.

[8] Cao, H., Jiang, D., Pei, J., Chen, E., and Li, H. 2009. Towards context-aware search by learning a very large variable length hidden Markov model from search logs. In Proc. WWW'2009, Madrid, Spain, 191-200.

[9] Castillo, C., Corsi, C., Donato, D., Ferragina, P., and Gionis, A. 2008. Query log mining for detecting spam. In Proc. Workshop on Adversarial information retrieval on the web, Beijing, China, 17-20.

[10] Chakrabarti, D., Agarwal, D., Josifovski, V. 2008. Contextual advertising by combining relevance with click feedback. In Proc. WWW'2008, Beijing China, 417-426.

[11] Chapelle, O., and Zhang, Y. 2009. A dynamic Bayesian network click model for web search ranking. In Proc. WWW-2009, Madrid, Spain.

[12] Craswell, N. and Szummer, M. 2007. Random walks on the click graph. In Proc. SIGIR'2007, Amsterdam, the Netherlands, 239-246.

[13] Craswell, N., Zoeter, O., Taylor, M., and Ramsey, W. 2008. An experimental comparison of click position-bias models. In Proc. WSDM'2008, Palo Alto, CA, 87-94.

[14] Cutrell, E., and Guan, Z. 2007. What are you looking for? An eye-tracking study of information usage in web search. In Proc. CHI'2007, San Jose, CA, 407-416.

[15] Dempster, A. P., Laird, N. M., and Rubin, D. B. 1977. Maximum likelihood estimation from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B, 39(1).

[16] Dupret, G. and Piwowarski, B. 2008. A user browsing model to predict search engine click data from past observations. In Proceeding SIGIR'2008, Singapore, 331-338.

[17] Fox, S., Karnawat, K., Mydland, M., Dumais, S., White, T. 2005. Evaluating implicit measures to improve web search. ACM Trans. on Information Systems, 23(3), April 2005, 147-168.

[18] Guan, Z. and Cutrell, E. 2007. An eye tracking study of the effect of target rank on web search. In Proc. CHI'2007, San Jose, CA, 417-420.

[19] Guo, F., Liu, C., Kannan, A., Minka, T., Taylor, M., Wang, Y.-M., and Faloutsos, C. 2009. Click chain model in web search. In Proc. WWW'2009, Madrid, Spain, 11-20.

[20] Huang, X., Acero, A., and Hon, H.-W. 2001. Spoken Language Processing, Prentice Hall, New Jersey.

[21] Joachims, T. 2002. Optimizing search engines using clickthrough data. In Proc. KDD'02, New York, 133-143.

[22] Joachims, T., Granka, L., Pan, B., Hembrook, H., Radlinski, F., and Gay, G. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. ACM Trans. on Information Systems, 25(2).

[23] Keane, M., O'Brien, M., and Smyth, B. 2008. Are people biased in their use of search engines? In Communications of ACM, 51(2), 49-52.

[24] Kelly, D., and Teevan, J. 2003. Implicit feedback for inferring user preference: a bibliography. SIGIR Forum, 37(2), 18-28.

[25] Klockner, K., Wirschum, N., and Jameson, A. 2004. Depth- and breadth-first processing of search result lists. In Proc. CHI' 2004, Vienna, Austria, 1539-1539.

[26] Liu, Y., Gao, B., Liu, T.-Y, Zhang, Y., Ma, Z., He, S. and Li, H. 2008. BrowseRank: letting web users vote for page importance. In Proc. SIGIR'2008, Singapore, 451-458.

[27] Lorigo, L., Haridasan, M., Brynjarsdottir, H., Xia, L., Joachims, T., Gay, G., Granka, L., Pellacini, F., and Pan, B. 2008. Eye tracking and online search: lessons learned and challenges ahead. J. of the American Society for Information Science and Technology, 59(7), 1041-1052.

[28] Pan, B., Hembrooke, H., Gay, G., Granka, L., Feusner, M., Newman, J. 2004. The determinants of web page viewing behavior: an eye-tracking study. In Proc. Symposium on Eye Tracking Research and Application (ETRA'04), 147-154.

[29] Radlinski, F., and Joachims, T. 2005. Query chains: learning to rank from implicit feedback. In Proc. KDD'2005, Chicago, IL, 139-148.

[30] Radlinski, F., Kurup, M., and Joachims, T. 2008. How does clickthrough data reflect retrieval quality? In Proc. CIKM'08, Napa Valley, CA.

[31] Rafter, R., and Smyth, B. 2001. Passive profiling from server logs in an online recruitment environment. In Proc. ITWP'01, Seattle, WA, 35-41.

[32] Richardson, M., Dominowska, E., and Rangno, R. 2007. Predicting clicks: estimating the clickthrough rate for new ads. In Proc. 16th International World Wide Web Conference (WWW'2007), Banff, Canada, 521-530.

[33] Terai, H., Saito, H., Egusa, Y., Takaku, M., Miwa, M., and Kando, N. 2008. Differences between informational and transactional tasks in information seeking on the web. In Proc. Information Interaction in Context (IIiX'08), London, UK, 152-159.

[34] Wang, K., Walker, T., and Zheng, Z. 2009. PSkip: Estimating relevance ranking quality from web search clickthrough data. In Proc. KDD'09, Paris, France.

[35] Wen, J.-R., Nie, J.-Y., and Zhang, H.-J. 2001. Clustering user queries of a search engine. In Proc. WWW'01, New York, NY, 162-168.

[36] White, R. and Drucker, S. 2007. Investigating behavioral variability in web search. In Proc. WWW'07, New York, NY, 21-30.