Efficient Multiple-Click Models in Web Search

Fan Guo*
Carnegie Mellon University
Pittsburgh, PA 15213
fanguo@cs.cmu.edu

Chao Liu
Microsoft Research
Redmond, WA 98052
chaoliu@microsoft.com

Yi-Min Wang
Microsoft Research
Redmond, WA 98052
ymwang@microsoft.com

ABSTRACT

Many tasks that leverage web search users' implicit feedback rely on a proper and unbiased interpretation of user clicks. Previous eye-tracking experiments and studies on explaining position-bias of user clicks provide a spectrum of hypotheses and models on how an average user examines and possibly clicks web documents returned by a search engine with respect to the submitted query. In this paper, we attempt to close the gap between previous work, which studied how to model a single click, and the reality that multiple clicks on web documents in a single result page are not uncommon. Specifically, we present two multiple-click models: the independent click model (ICM) which is reformulated from previous work, and the dependent click model (DCM) which takes into consideration dependencies between multiple clicks. Both models can be efficiently learned with linear time and space complexities. More importantly, they can be incrementally updated as new click logs flow in. These are well-demanded properties in reality.

We systematically evaluate the two models on click logs obtained in July 2008 from a major commercial search engine. The data set, after preprocessing, contains over 110 thousand distinct queries and 8.8 million query sessions. Extensive experimental studies demonstrate the gain of modeling multiple clicks and their dependencies. Finally, we note that since our experimental setup does not rely on tweaking search result rankings, it can be easily adopted by future studies.

Categories and Subject Descriptors

H.4 [**Information Search and Retrieval**]: Retrieval models

General Terms

Algorithms, Experimentation

*This work was done when the first author was on a summer internship with Microsoft Research.

This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version will appear in WSDM '09: Proceedings of the second ACM international conference on web search and data mining.
Copyright 2008 ACM.

Keywords

web search, click log analysis, statistical models

1. INTRODUCTION

The constantly growing web search traffic makes search activity logs a valuable source of information for understanding user preferences, which can be leveraged for many practical tasks including personalization (*e.g.*, [18, 20]) and optimization of search ranker (*e.g.*, [1, 11]). As one of the most direct and reliable evidence of user experience, clicks in the search result page can be easily logged and aggregated. One of the most challenging problems in click log analysis is to model the underlying mechanism that gives rise to clicks. Because of its importance, many researchers have investigated how to model user clicks (*e.g.*, [6, 12, 14]) and make use of developed models for downstream applications such as click log generation [9] and search engine evaluation [7]. A key trade-off in click model design is making reasonable user behavior assumptions for good predictive power, while preserving model efficiencies and robustness so that it can be applied to millions and billions of queries which is the typical in reality.

An important piece of work by Craswell et al. [6] proposed a *cascade model* and compared it with four basic models for user clicks. Specifically, the cascade model abstracts user clicks by taking two premise assumptions, both of which were supported by existing user studies ([12, 13]) and were adopted by other click model studies (*e.g.*, [9, 17]). The first assumption is the *examination hypothesis*, which states that a document must be examined before being clicked. It decouples the randomness of a click into a position-dependent probability of being viewed and a document-dependent probability of being clicked. This, to some extent, neutralizes the position-bias that is inherent in user clicks [14]. The second assumption is the *linear traversal hypothesis*, which suggests that users examine document abstracts by traversing the search result page from top to the bottom. Based on these two assumptions, the cascade model depicts how the *first* click arises: a web user examines documents one-by-one in order until the first click; after that the user never comes back. The capability of the cascade model is demonstrated with a controlled experiment based on real web users, which shows significant improvement over competing models. However, it assumes that a user abandons examination of web documents upon the first click. This unfortunately restricts the modeling power to query sessions with at most one click, which leaves the gap open for real-world applications where multiple clicks are possible, especially for in-

formational queries [4] which have a relatively large average number of clicks per query session.

Therefore, in this study, we investigate how to model query sessions with multiple clicks. We find that some previous work already shed lights on how to model multiple clicks, *e.g.*, [6, 7]. By gleaning their ideas we reformulate them as the independent click model (ICM), because it assumes (1) each web document in the search result is examined with probability one regardless of where it appears, and (2)click probabilities at different positions are independent. As our experiments suggest, despite its simplicity, ICM actually performs very well for some tasks, *e.g.*, predicting the first clicked position, but falls short for other challenging tasks like predicting where the last click happens.

Because clicks are inherently dependent, we propose to incorporate dependencies in modeling user clicks. We therefore propose the dependent click model (DCM) which generalizes the cascade model to multiple clicks by including a set of position-dependent parameters to model probabilities that an average user returns to the search result page and resumes the examination after a click. In return, it offers uniformly better performance than ICM, and does very well on the challenging task of predicting last clicked position.

Although DCM models dependencies between clicks as well as examination at different positions, it turns out to be efficient as well. Its worst time and space complexities are linear to the number of distinct web documents. Furthermore, DCM as well as ICM can be incrementally updated, which is a nice property for click log analysis in web search, considering click logs flow in as a data stream constantly. This will help propel its applications to handle the overwhelming web traffic nowadays. In comparison, a complicated model with sophisticated approximate inference algorithms generally does not scale well and incrementally update would be a non-trivial task for it.

In summary, we make the following contributions in this study:

- We investigate how to model user searches with multiple clicks, which closes the gap between previous work and reality. Specifically, two models (ICM and DCM) are put forward, with the former one gleaned from previous studies and the latter one developed by ourselves.
- We propose a set of experiments that can be easily adopted by future studies. The experiments presented in [6] for the cascade model, though good and convincing, is quite difficult to reproduce; it relies on tweaking the ranking algorithm in search engine implementation and presenting these results to real web search users.
- We present an extensive experimental study based on a data set containing over 8.8 million query sessions. Results demonstrate the gain of modeling multiple clicks and their dependencies in between.

The rest of this paper is organized as follows. We first elaborate on the two models in Section 2, and then present experimental studies in Section 3. Section 4 discusses related work, and this paper is concluded in Section 5.

2. MODEL

A web search user initializes a *query session* by submitting to the search engine a query string, or a *query*. Any resubmission or reformulation of the same query are regarded as distinct query sessions. The user may decide to click some or none of the web documents returned by the search engine. Search engine click logs then contain the submitted query, a ranked list of returned documents, whether each of them is clicked or not, and other information that might be useful. Here we focus on the first search result page, and discard any other elements in this page, *e.g.*, sponsored ads and related search. We use *document impression* to indicate appearance of web documents in search result pages at certain *positions*, or ranks. Documents in higher positions appear before those in lower positions.

Click models learn from user clicks to help understand and incorporate users' implicit feedback. In this paper, we follow a probabilistic approach which treats user clicks as random events, and the goal is to design generative models which are able to approximate underlying probabilities of clicks with high accuracy. Furthermore, we expect that distribution and statistics of samples generated from these models will match those of the empirical data closely.

For a given query, we assume that each document d is associated with a *document relevance* $r_d \in [0, 1]$, which is the probability that it is considered to be relevant to the query. Document impression in the query session of interest is denoted by $\{d_1, \ldots, d_M\}$, where M is the number of documents shown in the first page. Click models that adopt the examination hypothesis specify examination probabilities $e_{d_i,i} = \mathbb{P}(\text{examination at position } i \mid \text{document impression } d)$ for $1 \leq i \leq M$ as well as click probabilities $e_{d_i,i}$ defined in a similar fashion.

2.1 Modeling A Single Click

Single-click models are adopted from previous work in [6], which focused on modeling click position-bias. The simplest approach includes a universal click probability r, such that

$$c_{d_i,i} = r, \tag{1}$$

whereas a more reasonable assumption is that when there is no position-bias, clicks only depend on document impression:

$$c_{d_i,i} = r_{d_i}. \tag{2}$$

A more elaborate approach assumes that a user examines documents in the search result page one-by-one from the top position until the first click, and

$$c_{d_i,i} = r_{d_i} \prod_{j=1}^{i-1} (1 - r_{d_j}). \tag{3}$$

A click in position i implies that all positions above are skipped, *i.e.*, not clicked, and this click probability depends on the relevance of documents that appear at positions $1 \leq j < i$. This cascade model delivers better performance than a number of competing models overall, but unfortunately, works less favorably for bottom positions [6], which is mostly due to the single-click assumption.

2.2 Independent Click Model

Without positional-bias, click events at different positions are independent of each other. The single-click model specified in Eq. 2 can be directly applied to a whole query session, and this idea was mentioned in [6, 7]. Another way to derive this model is by assuming that users always come back after clicks in the cascade examination processes, and fitting multiple cascade models to the multiple clicks. Since

each position is examined with probability one, there is actually no position-bias and we obtain the same model with independent click probabilities. So here we reformulate this model as follows:

$$\begin{aligned} e_{d_i,i} &= 1, \\ c_{d_i,i} &= r_{d_i}, \end{aligned} \tag{4}$$

and name it the *independent click model* (ICM) in this paper. We will compare it with the model introduced next that considers dependent click probabilities in a query session.

Given the actual click event $\{C_1, \ldots, C_M\}$ in a query session as well as the document impression $\{d_1, \ldots, d_M\}$, the log-likelihood for ICM is given by

$$\ell_{ICM} = \sum_{i=1}^{M} \Big(C_i \log r_{d_i} + (1 - C_i) \log(1 - r_{d_i}) \Big). \tag{5}$$

Given a query and its corresponding query sessions in the training data, learning the ICM is to find r_d for every distinct document d to maximize the log-likelihood ℓ_{ICM}. The optimal value is achieved by simply setting r_d to the empirical click probability:

$$r_d = \frac{\#\text{ Click on } d}{\#\text{ Impression of } d}. \tag{6}$$

Therefore, we can set up two counting statistics to each document d, and parse only once through the training data to get all such counts, and finally compute all document relevance estimates. This leads to an learning algorithm with linear time complexity with respect to the number of query sessions and linear space complexity with respect to the number of distinct query-document pairs. When new data are available, we can do fast update and re-computation based on these counts, also in linear time and space complexities.

To evaluate multiple-click models in Section 3, we compute log-likelihood on the test data, and also draw samples from the ICM given document impressions for each query session in the test data:

$$\mathcal{C}_i \sim \text{Bernoulli}(r_{d_i}). \tag{7}$$

Then they are compared with the ground truth to obtain performance measures.

2.3 Dependent Click Model

In the cascade model a user always leaves the result page upon the first click and never come back. We propose to include a position-dependent parameter λ_i to reflect the chance that the user would like to see more results after a click at position i. In case of skip (no click), the next document is examined with probability one. λ_i's is a set of user behavior parameters shared over multiple query sessions. This user model is shown in Figure 1.

Examination and click probabilities in DCM can be specified in an iterative fashion ($1 \le i \le M$):

$$\begin{aligned} e_{d_1,1} &= 1, \\ c_{d_i,i} &= e_{d_i,i} r_{d_i}, \\ e_{d_{i+1},i+1} &= \lambda_i c_{d_i,i} + (e_{d_i,i} - c_{d_i,i}), \end{aligned} \tag{8}$$

from which the following closed-form equations can be derived:

$$e_{d_i,i} = \prod_{j=1}^{i-1} \left(1 - r_{d_j} + \lambda_j r_{d_j}\right), \tag{9}$$

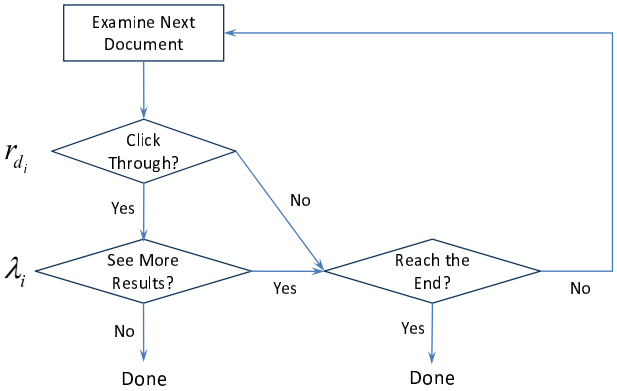

Figure 1: The user model of DCM, in which r_{d_i} is the document relevance of d_i, and λ_i is the user behavior parameter for position i.

$$c_{d_i,i} = r_{d_i} \prod_{j=1}^{i-1} \left(1 - r_{d_j} + \lambda_j r_{d_j}\right). \tag{10}$$

This completes the formal specification of the *dependent click model* (DCM), in which examine probabilities and click probabilities at different positions i become interdependent.

The log-likelihood for a query session with one or more clicks is given by

$$\begin{aligned} \ell_{DCM} = &\sum_{i=1}^{l-1} \Big(C_i(\log r_{d_i} + \log \lambda_i) + (1 - C_i) \log(1 - r_{d_i}) \Big) \\ &+ C_l \log r_{d_l} + (1 - C_l) \log(1 - r_{d_l}) \\ &+ \log \Big(1 - \lambda_l + \lambda_l \prod_{j=l+1}^{n} \left(1 - r_{d_j}\right) \Big), \end{aligned} \tag{11}$$

$$\begin{aligned} \ge &\sum_{i=1}^{l} \Big(C_i \log r_{d_i} + (1 - C_i) \log(1 - r_{d_i}) \Big) \\ &+ \sum_{i=1}^{l-1} C_i \log \lambda_i + \log(1 - \lambda_l). \end{aligned} \tag{12}$$

If there is no click in this session, then the log-likelihood is a special case with $l = M, C_l = \lambda_l = 0$.

We carry out DCM learning by maximizing the lower bound of log-likelihood in Eq. 12. Document relevance estimate for a document d is given by:

$$r_d = \frac{\#\text{ Click on } d}{\#\text{ Impression of } d \text{ before position } l}. \tag{13}$$

which is the empirical conditional click probability of d given it appears higher than or at position l. And the best estimate for the user behavior parameter

$$\lambda_i = 1 - \frac{\#\text{ query sessions when last clicked position} = i}{\#\text{ query sessions when position } i \text{ is clicked}}, \tag{14}$$

for $1 \le i \le M - 1$, which is the empirical probability of position i being a not-last-clicked position over all query sessions in the training set.

Compared with ICM, we only need additional $2(M - 1)$ global counts for λ_i's to carry out relevance estimate and parameter learning, which are still linear algorithms. Similar incremental updates are also applicable.

An important difference of DCM from ICM is that clicks indicate both relevance and examination. So if a document is not clicked, it can be attributed to either the document abstract is examined but not relevant enough to be clicked, or it appears lower than other documents that draw the user attention away. This explain-away effect is reflected in Eq. 13 by a smaller denominator which only counts impression before last clicks than that in Eq. 6 which counts every impression. However, for top-ranked documents that always appear before the last clicked position, the difference between two models is minor. This is consistent with the understanding of ICM as the model obtained by fitting multiple cascade models together in a straightforward way.

Finally, we give the sampling procedure for DCM which draws examination variables $\mathcal{E}_i$ and click variables $\mathcal{C}_i$ one-by-one starting from the top position:

$$\begin{aligned}
&\mathcal{E}_1 = 1; \\
&\text{If } \mathcal{E}_i = 0, \\
&\quad \mathcal{C}_i = 0,\ \mathcal{E}_{i+1} = 0; \\
&\text{else} \\
&\quad \mathcal{C}_i \sim \text{Bernoulli}(r_{d_i}), \\
&\quad \mathcal{E}_{i+1} \sim \text{Bernoulli}(1 - \mathcal{C}_i + \lambda_i \mathcal{C}_i).
\end{aligned} \tag{15}$$

3. EXPERIMENT

We report our experimental studies in this section, which is based on over 8.8 million queries sessions after data preprocessing, sampled from the click log of a major commercial search engine in July 2008. Experimental results indicate that both multiple-click models fit the reality well: 8 times better log-likelihood results than a baseline approach. Especially, DCM offers much better last clicked position prediction than ICM and more reasonable click distributions by virtue of its modeling assumption that clicks at different positions are interdependent. In the following, we start with experimental setup in Section 3.1, and proceed with detailed results in Sec. 3.2, and finally conclude with a summary of experimental findings in Section 3.3.

3.1 Experiment Setup

The data set is obtained by sampling the click log of a major commercial search engine during July 2008. The click log consists of the query string, the time-stamp, document impression data (URLs of top-10 documents in the first page) and click data (whether each document is clicked or not) for each query session. Only query sessions with at least one click are kept for better data quality since we find from additional meta-information that clicks on ads, query suggestions or other elements are much more likely to appear for the ignored sessions with no clicks. It also provides clearer comparison of performances on predicting the first and last clicked position. For each query, we sort its query sessions by time-stamp and split them into training set and test set of equal sizes. The number of query sessions in the training set is 4,804,633. Then these queries are categorized according to the query frequency in the test set. Top 0.16% (178) most frequently searched queries (also known as *head queries*) with frequencies greater than $10^{3.5}$ are not included in the subsequent results on test set because most search engines already do very well on these queries. After data preprocessing, the test set consists of 4,028,209 query sessions for 110,630 distinct queries in 6 query frequency categories. The average number of clicks per query session is 1.139.

Table 1: Summary of Test Set

Query Freq	# queries	# Sessions	Avg # Click
1∼9	59,442	216,653 (5.4%)	1.310
10∼31	30,980	543,537 (13.5%)	1.239
32∼99	13,667	731,972 (17.7%)	1.169
100∼316	4,465	759,661 (18.9%)	1.125
317∼999	1,523	811,331 (20.1%)	1.093
1000∼3162	553	965,055 (24.0%)	1.072

Statistics of the test set are summarized in Table 1. Note that our data set includes a great number of tail queries which are often ignored in experiments conducted in previous studies, and performances over all query sessions are not dominated by head queries or a particular query frequency range.

For each query, document relevance estimates are computed using Eq. 6 and Eq. 13 for ICM and DCM respectively on the training data. But for documents which appear very few times in the training set and which appear only in the test set, document relevance are replaced by position relevance, which are computed for each position in a similar way, for deriving log-likelihood and other metrics in the test set. This has a smoothing effect on the document relevance, and leads to better performance for the evaluation on the test data. Since the additional counts that we need to keep in the computation, $2M$ for each query, is usually much smaller than the cost saving from low-frequency documents, the time and space complexities can also be reduced. The cut off of minimum number of impression for document relevance computation is set adaptively according to the query frequency category from 1 to 6. Finally, to avoid infinite values in the evaluation, we further imposes a lower bound of 0.01 and an upper bound of 0.99 on the learned relevance values for both models as well as user behavior parameters in DCM.

Parsing the data from the hard disk and loading them into main memory takes around 45 minutes. All the subsequent experiments are carried out in a server machine, with 2.67GHz CPU cores, 32GB memory, Windows Server 2008 64-bit OS, and MATLAB R2008a installed. The computational time for training DCM is no more than 7 minutes.

3.2 Experimental Evaluation

3.2.1 Results on Log-Likelihood

Figure 2 presents log-likelihood curves for different query frequencies, where larger log-likelihood results indicate better fit on the test data. Besides DCM and ICM, the baseline model specified in Eq. 1 is also implemented. DCM achieves larger performance gain for more frequent queries, and consistently outperforms ICM by over 10% when the query frequency is over 100. Both DCM and ICM have over 8 times larger likelihood than the baseline; the difference is only less significant for tail queries of frequencies less than 10.

The DCM curve goes below ICM for queries with frequencies less than $10^{1.5}$. But this does not imply that we should always apply ICM to model these queries. Instead, we suggest that lower confidence should be given in document relevance estimates derived from click models for these tail queries. We could still record counting statistics for these queries, but document relevance estimates should be reliable

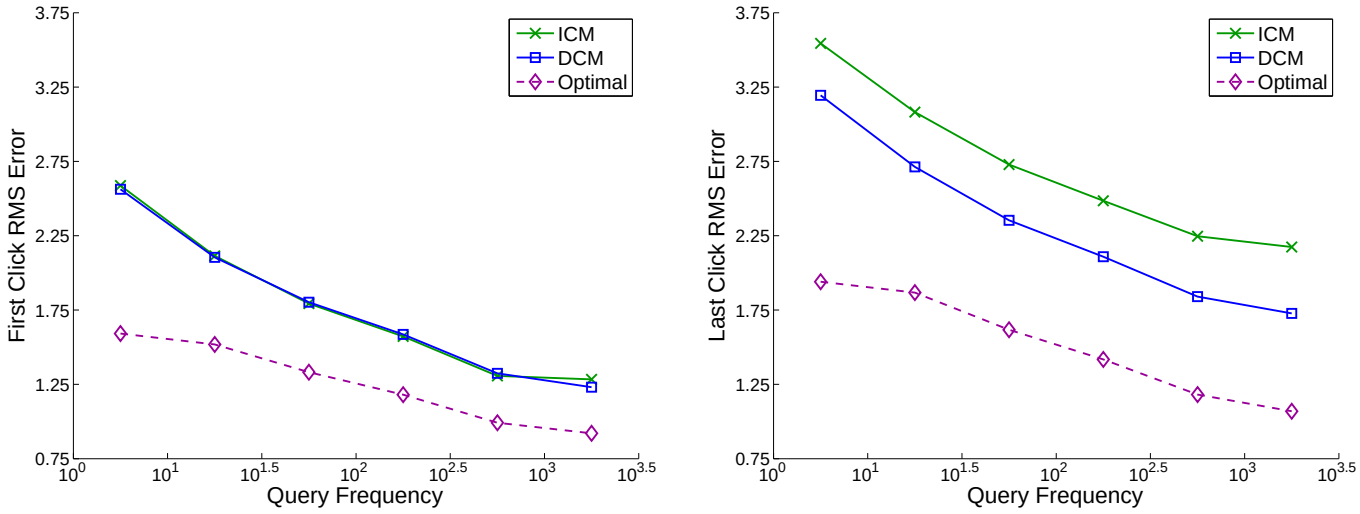

Figure 3: Root-mean-square (RMS) errors for predicting first clicked position (a) and last clicked position (b). Results are averaged over 100 samples per query session.

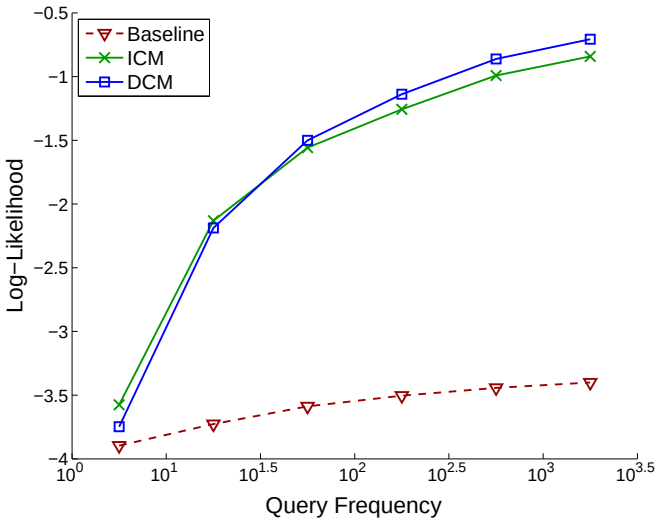

Figure 2: Log-likelihood per query session on the test data for different query frequencies. The overall average for DCM is -1.327, compared with -1.401 for ICM and -3.534 for the baseline method.

when new data flow in and the amount of training data is enough to obtain a good fit.

3.2.2 *Predicting First and Last Clicks*

We now focus on clicks and test whether samples generated from ICM and DCM provide good match of first and last clicked position compared with the empirical data. Given each query session in the test set, we use the document relevance learned from the training set to determine the click probability. For ICM, clicks are sampled for each position independently, whereas for DCM, sampling starts from the top ranked document and ends at either the first non-examined position or the last (10th) position. For both models, we collect 100 samples with at least one click, then first and last clicked position are identified from the simulated click data and compared with the ground truth to compute RMS errors. This is the most time-consuming part in the model evaluation experiments and takes around one hour to finish. To reflect the inherent randomness in user click behavior, we also compute for each query the standard deviation of first and last clicked position and take a weighted mean over different queries to approximate the lower bound of RMS error. This corresponds to the "optimal" curves in Figure 3. We expect a model that gives consistently best fit of click data would have the smallest margin with respect to the optimal error, and this margin also reflects the robustness of model prediction since the RMS error metric takes account of both bias and variance in prediction. Finally, we aggregate results over all queries and compare the distribution of first and last clicks from two click models with the empirical distribution of the test data, which corresponds to the "empirical" curves in Figure 4.

RMS errors for ICM and DCM are close for first clicked position because their model assumptions are the same until the first click. Predicting last clicked position turns out to be a more difficult task as demonstrated by higher error curves in Figure 3(b) than 3(a). With a position-dependent modeling assumption, DCM outputs more reasonable last click estimates than ICM, reducing the RMS error gap from the optimal curve by around 30%.

Figure 4 illustrates generally slower than geometric decrease with the position for the empirical probabilities of both first and last clicks. DCM matches these probabilities very well at the top 5 positions. The higher tail of empirical curves is probably due to user scrolling behaviors, especially for informational queries which have a higher click through rate. And we suspect that users may examine documents in a different fashion when they scroll to the bottom of the search result page, so that the 10th position receives even more last clicks than the two above. However, they contribute to a fairly small fraction of overall results: clicks after position 6 represent only 6.1% of the total number. For

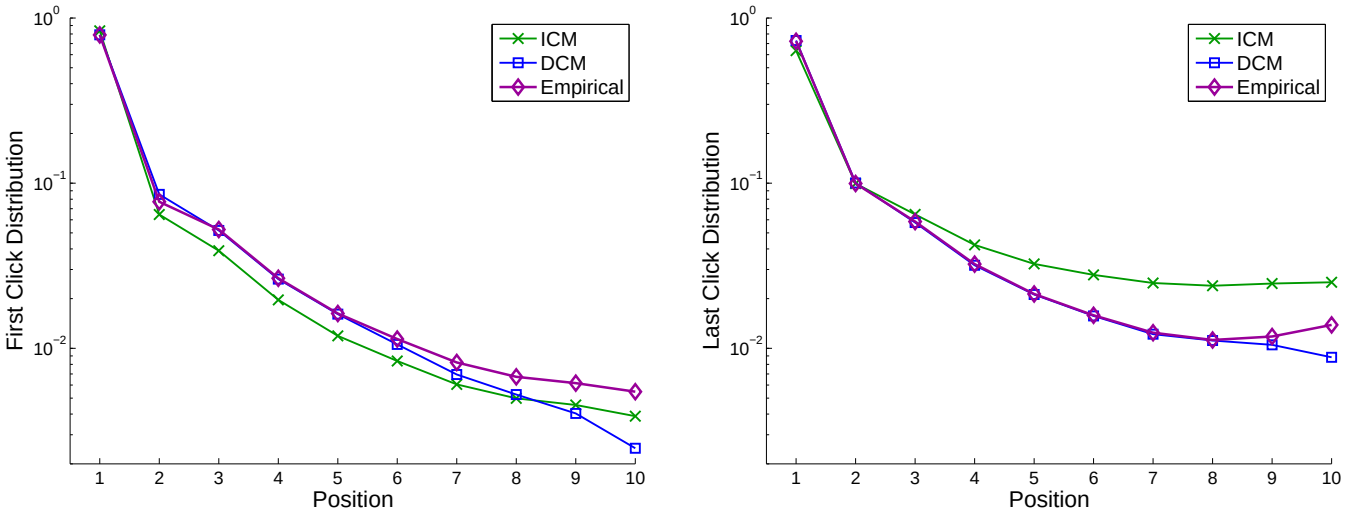

Figure 4: First click distribution (a) and last click distribution (b) obtained by drawing samples from DCM and ICM given document impression. The overall first/last click distribution of DCM samples matches the empirical distribution in the test set very well, particularly for top 5 positions. Results are averaged over 100 samples per query session.

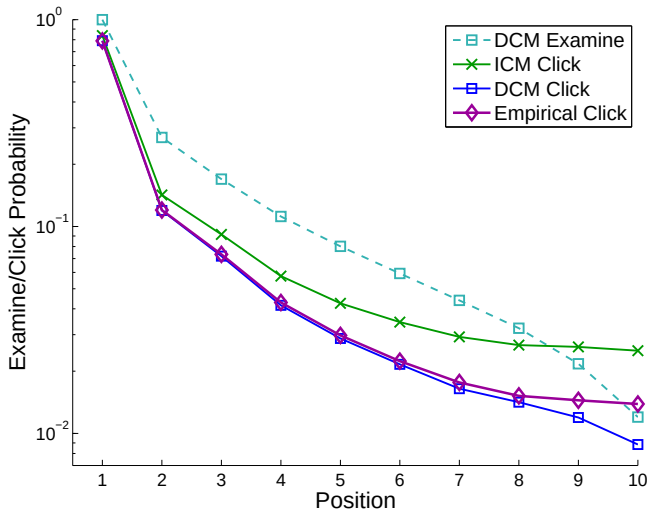

Figure 5: Click probabilities for different positions summarized from ICM/DCM samples as well as test data, and examine probabilities implied by DCM. The click distribution implied by DCM matches the ground truth closely.

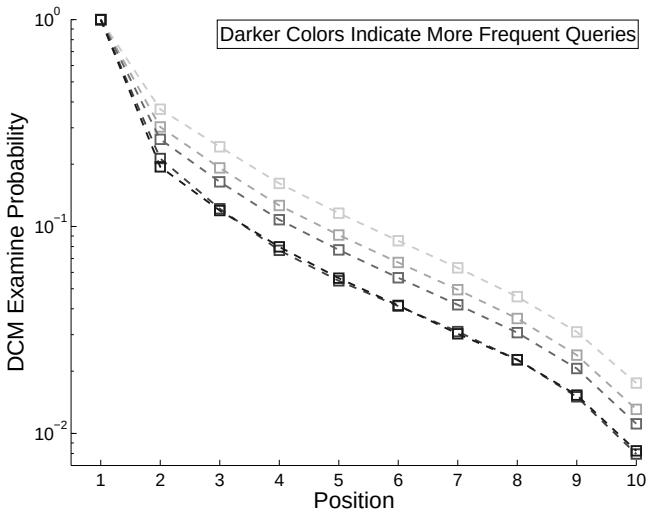

Figure 6: Examine probabilities implied by DCM for different query frequencies. Queries are grouped into 6 frequency ranges similarly as in Table 1. Darker and lower curves correspond to more frequent queries.

ICM samples, documents that appears in lower positions may receive more clicks than the ground truth because of the position-independent assumption. This results in over-estimation of last click probabilities for these positions in Figure 4(b). On the other hand, the document relevance estimates in ICM is smaller than those in DCM, due to a larger denominator in computing the empirical probabilities. This under-estimation has a more significant effect on documents which usually appear in lower positions and after the last clicked position. Therefore, the first click probability distribution derived from ICM has a lower tail than the empirical curve, as shown in Figure 4(a).

3.2.3 *Examination and Click Curves*

A unique property of DCM is that examination probabilities could be computed for each query session and they are aggregated together to provide a hint on user attention over different positions, which corresponds to the dashed curve in Figure 5. The first position is always examined from the modeling assumption, followed by a geometrically decreas-

ing pattern after position 2. Compared with the DCM click curve, the gap between them reflects the log conditional click probabilities for each position, which suggests larger probabilities for both top and bottom positions. Note that both curves go below the empirical click for the last position, and this bias is attributed to user behaviors beyond the modeling assumption as discussed for first and last click distributions. Moreover, putting the empirical click and last click probabilities together, it seems that a number of users tend to click the bottom few positions simultaneously, leading to an slightly upward tail in Figure 4(b) for empirical curves.

Figure 6 displays detailed examination probabilities for different query frequencies. All of them share similar decreasing pattern but differ in absolute values. The trend is that less frequent queries tend to be examined in greater depth, and we also observe more clicks per query session in the click log for them.

3.3 Experiment Summary

The extensive experimental study carried out in this study has demonstrated the gain of modeling multiple clicks and their dependencies. Specifically, both multiple-click models achieve 8 times larger likelihood on the test data than a baseline approach, and DCM is 7% percent better than ICM. Despite its simplicity, ICM actually performs very well for predicting the first clicked position, but falls short for the challenging task of predicting where the last click happens compared with DCM (15% margin) which effectively captures click dependencies. Click distributions derived from DCM matches empirical ground truth on the test data very well, especially for the top 5 positions which receive over 90% of total clicks. Finally, it is interesting to point out the fact that the category of most frequent queries, which has no more than 8 percent of multiple-click query sessions, benefits most from the multiple-click model. So it is the amount of data per query that plays an important role in boosting the performance of click models.

4. RELATED WORK

Previous work most relevant to this paper is the cascade model presented in [6], earlier work carried out by Dupret et al. [7, 9], and eye tracking studies in [12, 13]. They have been discussed in Section 1.

Joachims [11] presented a pioneering study to exploit clickthrough data for optimizing the ranking function for search engines. Pairwise preference feedback, such as web document i is more relevant as web document j, are extracted from click logs and used to train a ranking support vector machine (ranking SVM) to output a retrieval function most concordant with these partial orderings. It was extended by Radlinski et al. [15], and an algorithm was proposed to detect a sequence of reformulated queries from the same user to learn an improved function. Radlinski et al. [16] followed this line of study for optimizing ranking functions but takes an alternative active-learning approach to control documents presented to users in search result pages for obtaining more helpful feedback as the next-round training data. The approach we take in this paper is different from these previous studies in that clicks are treated as random events under an explicit user model and document relevance is interpreted as click probabilities upon examination.

Clickthrough data could be also combined with other implicit measures or browsing data available from query logs to improve web search. The studies by Agichtein et al. proposed to extract a spectrum of features from browsing and click activities as well as textual data to train a better ranker [1] and estimate user preference [2]. An earlier work in evaluating these implicit measures appeared in [10]. Note that these additional information may not be able to be collected everyday due to the huge search volume. And it may also be subject to high level of noise, *e.g.*, web page dwelling time may be inaccurate if a user locks the screen to have a break with the browser open.

One of the earliest publications on large scale query log analysis [19] appeared in 1999 which presented interesting statistics as well as a simple correlation analysis from the Alta Vista search engine. Xue et al. [21] proposed to use clickthrough data to improve graph-based static ranking algorithms. Bilenko et al. [3] presented a novel study in identifying "search trails" from user activity logs and used a random-walk based algorithm for improved retrieval accuracy. In [5] Carterette et al. proposed a logistic model to predict document relevance using scores obtained from human judges.

There is a very recent paper on click models by Dupret and Piwowarski [8] in SIGIR'08, but we were limited by time to implement their models and algorithms on our data set and compare with the DCM we presented. So here we give a brief discussion from a theoretical point of view on these two independent threads of click model studies. Both DCM and user browsing models proposed in [8] adopt the examination hypothesis and the linear traversal hypothesis, but they differ in the specification of examination probabilities over different positions. In DCM, they are derived from a first-order Markovian examination processes. In user browsing models, the specification is more complicated and involves $M(M+1)/2$ user behavior parameters compared with $(M-1)$ for DCM. Accordingly, parameter learning and document relevance estimation cost more time and space in user browsing models. They are performed by an iterative algorithm under the coordinate ascent framework in [8], which requires multiple scans through the training data and each scan is an order of magnitude more expensive than DCM. Incremental updates for user browsing models would also be a non-trivial task. DCM, on the other hand, features a simpler specification and more efficient algorithms, but this does *not* necessarily imply less effective or less robust performance. Due to the differences in experimental setting as well as the data set property between [8] and this study, the reported results are not directly comparable. However, it is interesting to point out that in [8] the evaluation metrics is the perplexity, which focuses on prediction accuracy at each position individually, whereas evaluation metrics in this paper, in particular log-likelihood and last clicked position prediction, are based on click prediction over different positions in a query session. To achieve good performance in these two metrics, a model has to be able to capture click dependencies between different positions.

5. CONCLUSION

We attempt to close the gap between previous work and the reality for modeling query sessions with multiple clicks, by presenting two click models: ICM and DCM. Theoretical derivation shows that both models have linear time and space complexities and a desired property which allows fast incremental computation. Extensive experimental stud-

ies demonstrate that DCM makes good trade-off between model complexity and efficiency, and offers better performance than ICM. Performance gain is most significant on log-likelihood and predicting last clicked position, where the ability to model click dependencies is a must. Furthermore, our experimental setup does not rely on tweaking search engine algorithms or active user participation, so it can be easily adopted or reproduced by future studies. Our future work includes the design of more robust click models and comparing the performance of existing click models.

6. ACKNOWLEDGMENTS

We would like to thank Nick Craswell for the discussion and comments, and the reviewers for suggestions that improved the presentation. We are also grateful to Ethan Tu and Li-Wei He for their help. Thanks to Kaisen Lin and Alex Rasmussen for their comments on the manuscript.

7. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26, 2006.

[2] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10, 2006.

[3] M. Bilenko and R. W. White. Mining the search trails of surfing crowds: identifying relevant websites from user activity. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 51–60, New York, NY, USA, 2008. ACM.

[4] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

[5] B. Carterette and R. Jones. Evaluating search engines by modeling the relationship between relevance and clicks. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 217–224. 2008.

[6] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *WSDM '08: Proceedings of the first ACM international conference on Web search and data mining*, pages 87–94, 2008.

[7] G. E. Dupret, V. Murdock, and B. Piwowarski. Web search engine evaluation using click-through data and a user model. In *Proceeding of the Workshop on Query Log Analysis: Social and Technological Challenges (WWW '07)*, 2007.

[8] G. E. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 331–338, 2008.

[9] G. E. Dupret, B. Piwowarski, C. A. Hurtado, and M. Mendoza. A statistical model of query log generation. In *String Processing and Information Retrieval, 13th International Conference, SPIRE 2006*, pages 217–228, 2006.

[10] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Trans. Inf. Syst.*, 23(2):147–168, 2005.

[11] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, 2002.

[12] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, 2005.

[13] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.*, 25(2):7, 2007.

[14] T. Joachims and F. Radlinski. Search engines that learn from implicit feedback. *Computer*, 40(8):34–40, 2007.

[15] F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 239–248, 2005.

[16] F. Radlinski and T. Joachims. Active exploration for learning rankings from clickthrough data. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 570–579, 2007.

[17] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 521–530, 2007.

[18] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50, 2005.

[19] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.

[20] B. Tan, X. Shen, and C. Zhai. Mining long-term search history to improve search accuracy. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 718–723, 2006.

[21] G.-R. Xue, H.-J. Zeng, Z. Chen, Y. Yu, W.-Y. Ma, W. Xi, and W. Fan. Optimizing web search using web click-through data. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 118–126, 2004.