Tailoring Click Models to User Goals

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

Lei Li
Carnegie Mellon University
Pittsburgh, PA 15213
leili@cs.cmu.edu

Christos Faloutsos
Carnegie Mellon University
Pittsburgh, PA 15213
christos@cs.cmu.edu

ABSTRACT

Click models provide a principled way of understanding user interaction with web search results in a query session and a statistical tool for leveraging search engine click logs to analyze and improve user experience. An important component in all existing click models is the user behavior assumption – how users scan, examine and click web documents listed in the result page. Usually the average user behavior pattern is summarized in a small set of global parameters. Can we fit multiple models with different user behavior parameters on a click data set? A previous study showed that the mixture modeling approach did not lead to better performance despite extra computational cost.

In this paper, we present how to tailor click models to user goals in web search through query term classification. We demonstrate that better predicative power could be achieved by fitting two click models for navigational queries and informational queries respectively, as evidenced by the likelihood and perplexity evaluation results on a subset of the MSN 2006 RFP data which consists of 121,179 distinct query terms and over 2.8 million query sessions. We also propose search relevance score (SRS) as a flexible evaluation metric of search engine performance. This metric can be derived as summary statistics under any click model, and is applicable to a single query session, a particular query term and the search engine overall.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*relevance feedback, retrieval models*

General Terms

Algorithms, Experimentation, Measurement

Keywords

Web search, click model, user behavior

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
WSCD '09, Feb 9, 2009 Barcelona, Spain.
Copyright 2009 ACM 978-1-60558-434-8 ...$5.00.

1. INTRODUCTION

Researchers sought to understand user browsing and navigation behaviors well before the dot-com era of the web, usually through personal interview and questionnaire-based survey (*e.g.*, [12]). The first large-scale analysis of search engine query logs to our knowledge dates back to 10 years ago, carried out by Silverstein *et al.* [16], using over 285 million sessions collected from all the search traffic of AltaVista during 43 days in August and September 1998. What is missing in this early study is the click data – among the ranked list of search results which ones are clicked through.

Click data is now one of the most important and extensive feedback signals from the WWW audience. And a number of approaches have been proposed to leverage click logs to improve the user experience of web search engines. For example, Joachims [8] put forward an SVM algorithm to optimize the ranking function using pairwise relevance judgement extracted from clickthrough as the input. A clicked web document is considered more relevant to the query term than a skipped one that appears above. Agichtein *et al.* [1] explored a number of alternatives for incorporating user behavior features into popular web search rankers such as RankNet [3]. Click data have also been applied to the evaluation of search engine performance [4, 13].

However, the interpretation of user clicks is a non-trivial task because many elements come into play in this decision process. Previous eyetracking studies [9, 10] indicated that clicks are biased as a form of absolute relevance judgement, and clicking decision on a web document depends on both the position (rank) and the context (other documents) of the presentation. Richardson *et al.* [14] suggested the *examination hypothesis* as a general solution to account for the position bias, under which the chance of a click is decoupled to two factors: the examination probability and the document relevance. For example, a document that is ranked at the bottom of the search result may not be clicked simply because few users would pay attention to it. Recently, Craswell *et al.* [5] proposed the *cascade hypothesis* which assumes that users sequentially scan each document in the list of search results until reaching the first click, and the clicking decision on the document is made after examination and before going to the next one in the list.

Click models provide a principled way of understanding user interaction with web search results in a query session. They usually incorporate user behavior assumptions, such as the examination hypothesis and the cascade hypothesis, to specify how examination and clicks at different positions depend on each other. Given the click log which includes the

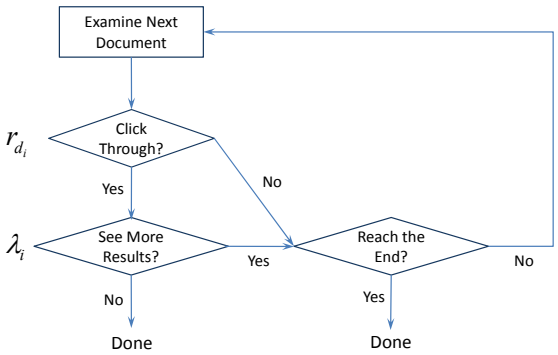

Figure 1: The user model of DCM, in which r_{d_i} is the document relevance of d_i, and λ_i is global parameter for position i.

query term, a ranked list of web documents and click data for each query session, model learning outputs document relevance estimation with respect to the query term, as well as a set of model parameters characterizing the underlying user behavior. The estimation and learning algorithms have to be efficient and incremental to be applied on real world scenario with millions of new query sessions every day.

A common assumption in existing click models [6, 7] is that query sessions are independent of each other and share the same user behavior model, while in the real world users may have different browsing strategies for different query terms. In [6] a mixture-modeling approach was proposed so that query terms could be clustered softly using an EM-based iterative algorithm. However, despite the expensive computational cost and extra algorithmic difficulty for incremental update, fitting multiple models failed to provide better performances than a single click model. On the other hand, the authors were aware of two general categories of query terms in existing literatures [2, 15]. Navigation queries are generated when a user has a particular web site in mind and only need to find a link to the final destination, whereas the purpose of informational queries is to obtain information about the query content. This difference in user goals leads to different browsing and click patterns. And based on features derived from click data, query terms could be classified with good accuracy as demonstrated by Lee *et al.* [11].

In this paper, we present a simple, yet effective approach of tailoring click models to user goals. We adapt the idea in [11] to classify query terms, and demonstrate that with little extra computation, fitting two click models achieves significant gain in the model predicative power. Moreover, our approach offers the first quantitative comparison of the difference between user examination behavior for navigational and information queries, as well as other summary statistics derived from the click models learned. In particular, we propose the *search relevance score* as a flexible evaluation metric of search engine performance, which is applicable to different granularity levels such as a single query session, a particular query term and the search engine overall.

The remainder of the paper is organized as follows. In Sec. 2 we briefly introduce the recently proposed dependent click model as the running example, and present algorithms for parameter estimation and user goal identification. In Sec. 3 we show how to derive the search relevance score. Experimental evaluation is covered in Sec. 4, and the paper is concluded in Sec. 5.

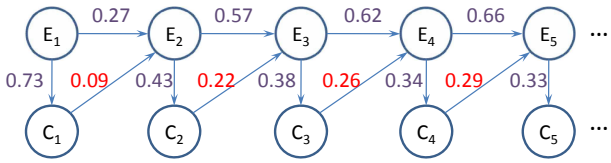

Figure 2: The first half of the state transition diagram of DCM. Numbers shown are the average probabilities learned on the experimental data.

2. MODEL AND ALGORITHMS

The notation in this paper is similar to [7]. A *query session* is initialized by a web search user when the *query* (or *query term*) is submitted to the search engine. Query resubmission and reformulation is treated as independent query sessions. Only the organic search results without ads, query suggestions shown on the first page are considered as *web documents* in the model. They appear in a ranked list where a document in a higher *position* is followed by those in lower positions. For a particular query session, documents in the ranked list are denoted by $\{d_1, \ldots, d_M\}$, where M is the length of the list. Each document d_i is associated with a *document relevance* $r_{d_i} \in [0, 1]$. Click models specify examination probabilities $e_{d_i,i}$ and click probabilities $c_{d_i,i}$ for each position i.

2.1 Dependent Click Model

The user behavior assumption of dependent click model is depicted in Fig. 1. A user always examines the first position

$$e_{d_1,1} = 1, \tag{1}$$

and the probability of a click upon examination always equals the document relevance as in the examination hypothesis

$$c_{d_i,i} = e_{d_i,i} r_i \qquad (1 \le i \le M). \tag{2}$$

If there is no click at a position $i < M$, the user always continues to examine the next document d_{i+1} as in the cascade hypothesis. The probability of resuming the examination after a click is λ_i:

$$e_{d_{i+1},i+1} = (e_{d_i,i} - c_{d_i,i}) + c_{d_i,i}\lambda_i \qquad (1 \le i < M). \tag{3}$$

The user behavior parameters in DCM are therefore $\lambda_1, \ldots, \lambda_{M-1}$. They are shared by all query sessions under the model.

The model can also be illustrated using a state-transition diagram in Fig. 2. E_i and C_i correspond to the state of examination and click at position i respectively. The initial state is E_1 and there is an absorbing state representing the end of the query session (not shown in the figure).

2.2 Estimation Algorithm

We follow the general procedure of maximum-likelihood estimation. The log-likelihood for a query session whose last clicked position is l

$$\ell = \sum_{i=1}^{l-1} \Big(C_i(\log r_{d_i} + \log \lambda_i) + (1 - C_i)\log(1 - r_{d_i}) \Big) + C_l \log r_{d_l} + \log \Big(1 - \lambda_l + \lambda_l \prod_{j=l+1}^{n} (1 - r_{d_j}) \Big), \tag{4}$$

$$\geq \sum_{i=1}^{l} \Big(C_i \log r_{d_i} + (1 - C_i) \log(1 - r_{d_i}) \Big) + \sum_{i=1}^{l-1} C_i \log \lambda_i + \log(1 - \lambda_l). \quad (5)$$

If there is no click in this session, then the log-likelihood is simply $\ell = \sum_{i=1}^{M} \log(1 - r_{d_i})$.

Here we introduce an efficient learning algorithm which maximizes the lower bound of log-likelihood in Eq. 5. Given a query term, we keep the following three counts for each of its document d:

- C_1^d: the number of query sessions d appears before the last clicked position and is not clicked;
- C_2^d: the number of query sessions d appears before the last clicked position and is clicked;
- C_3^d: the number of query sessions d is the last clicked document.

Then the document relevance estimate is

$$r_d = \frac{C_2^d + C_3^d}{C_1^d + C_2^d + C_3^d}. \quad (6)$$

For more robust estimation, we add 1 to the numerator and add 2 to the denominator for each document as pseudo-counts to provide some smoothing.

Similarly we keep three global counts for each position i

$$\lambda_i = \frac{C_2^i}{C_2^i + C_3^i} \qquad (1 \leq i < M) \quad (7)$$

Therefore we need to keep 3 counts for each query-document pair of interest, and a total of $3M$ counts for estimating the user behavior parameters. To collect these counting statistics, we only need a single pass through the log data. And when new logs flow in, we similarly update the counts and create additional ones for every new query-document pair when necessary.

2.3 Identifying User Goals

To classify each query term into one of the two general categories of navigational queries and informational queries, we first compute the number of clicks on each position and sort them in descending order to obtain a click vector $C = \{c_1, \ldots, c_M\}$. This vector characterizes how user clicks distribute over positions. We also record the time elapsed in the query session before the first click happens. Following a similar approach as in [11], we can derive a single numeric feature based on these statistics and define a cut-off threshold. In particular, we propose and implement the following metrics:

- *MeanClk*: $\frac{\sum_{i=1}^{M} i c_i}{\sum_{i=1}^{M} c_i}$, the mean of click distribution.
- *MedClk*: $\min\{m | \sum_{i \leq m} c_i > \sum_{i > m} c_i\}$, the median of click distribution.
- *AvgClk*: the average number of clicks per query session.
- *MedTime*: the median of time (in second) spent before the first click.

For every metric above, if the value is less or equal than the threshold, then the query term is identified as a navigational query, otherwise, it belongs to the set of informational queries. The cut-off can be set empirically to optimize the evaluation score on the training set.

After we have identified the user goals for each query, we are ready to fit two DCMs on the click log. The only change that need to be made is that we have to keep two sets of global counts for estimating user behavior parameters, and update the counts according to the type of query term. When we compute the log-likelihood and other summary statistics on the test set, our choice of user behavior parameters also depend on whether the query is navigational or informational.

3. SEARCH RELEVANCE SCORE

Given the learned document relevance and user behavior parameters, the examination probabilities for each position ($1 \leq i \leq M$) can be derived from Eqs. 1∼3 in Sec. 2.1:

$$e_{d_i,i} = \prod_{j=1}^{i-1} \big(1 - r_{d_j} + \lambda_j r_{d_j}\big). \quad (8)$$

The search relevance score (SRS) is defined as the expected examined document relevance, *i.e.*, the average document relevance weighted by the examination probabilities:

$$SRS = \frac{\sum_{i=1}^{M} e_{d_i,i} r_i}{\sum_{i=1}^{M} e_{d_i,i}} = \frac{\sum_{i=1}^{M} c_{d_i,i}}{\sum_{i=1}^{M} e_{d_i,i}} \quad (9)$$

The computation above obtains the SRS for a particular query session. The score can also be aggregated for each query term by adding together the numerators (click probabilities) and denominators (examination probabilities) in Eq. 9 for all its query sessions respectively and doing a division. Similar aggregation procedure could be applied over all query terms to obtain the search engine score.

Other interesting summary statistics can also computed under DCM, for example, the expected last examined position, also known as examination depth, is given by

$$\text{Examination Depth} = \frac{\sum_{i=1}^{M-1} i e_{d_i,i} r_{d_i} (1 - \lambda_i) + M e_{d_M,M}}{\sum_{i=1}^{M-1} e_{d_i,i} r_{d_i} (1 - \lambda_i) + e_{d_M,M}} \quad (10)$$

4. EXPERIMENT

The data set comes from the MSN 2006 RFP data. Usually each search result page contains 10 documents. Query sessions with clicks after the 10th position and with less than 10 documents listed are discarded. Also, only query sessions with at least one click are kept for better data quality since clicks on ads and query suggestions are not logged in the data. For each query term, we order its query sessions by time and split them equally into the training set and the test set. The query frequency has to be at least 3 in both sets. After these preprocessing, there are 121,179 distinct query terms. The training set contains 1.52 million query sessions, and the test set contains 1.21 million query sessions, with statistics summarized in Table 1.

In this query log, we don't know the identity of the document when it is in the search result and it is not clicked, therefore we only compute the relevance of each position as

Table 1: A Summary of the Test Data Set.

Query Freq	# Terms	# Sessions	Avg # Click
3∼9	74,649	329,567 (25.3%)	1.281
10∼31	15,072	242,073 (18.6%)	1.233
32∼99	3,871	203,805 (15.7%)	1.175
100∼316	1,072	179,975 (13.8%)	1.117
317∼999	298	156,839 (12.1%)	1.091
≥1000	83	188,169 (14.5%)	1.064

Table 2: Comparison of Different Approaches of Tailoring Click Models to User Goals. Percentage of navigational queries is shown in the second column.

Approach	Cut-off	LL	Perplexity
MedClk	1	**-1.6804(4.0%)**	**1.2303(1.3%)**
MeanClk	2.0	-1.6826(3.7%)	1.2306(1.2%)
AvgNClk	1.2	-1.6901(3.0%)	1.2308(1.1%)
MedTime	6.0	-1.6948(2.5%)	1.2315(0.8%)
Baseline	N/A	-1.7192	1.2333

well as the user behavior parameters during model training. The running time for model training is 150 seconds on a UNIX server with eight 3.0GHz cores and 16GB main memory.

We compare the average log-likelihood and perplexity results on the test data to measure model fitness. Log-likelihood (LL) is defined as the log probability of observed click events under the learned click model. The optimal LL value is 0 and the improvement of ℓ_1 over ℓ_2 is reported as $(\exp(\ell_1 - \ell_2) - 1) \times 100\%$. Perplexity is defined to capture click prediction quality for each position in a query session independently. Perplexity $p_i = 2^{-(C_i \log_2 q_i + (1-C_i)\log_2(1-q_i))}$ if C_i is the actual binary click event as position i and q_i is the model prediction. The optimal value is 1 and the improvement of perplexity value p_1 over p_2 is reported as $(p_2 - p_1)/(p_2 - 1) \times 100\%$. Note that average perplexity values are computed using geometric mean since it is not in log scale and smaller perplexity value indicates higher prediction quality.

Table 2 lists the evaluation results. Median of click distribution, as a robust statistics, is the best measure in fitting two click models, and leads to a 4% improvement over the baseline in log-likelihood, as well as a 1.3% better perplexity values. The larger margin in log-likelihood is expected because multiple models could better capture click dependencies for heterogenous user browsing and click patterns, whereas the perplexity for individual positions, especially the top few, will be less sensitive.

Figure 3 depicts the clear difference in the examination and click distribution derived by the two models under the MedClk approach. Probability curves of the navigational model are much steeper than its informational counterparts, and the decreasing rates are largest at top positions, whereas the click probability curve of the informational model imply a close-to-exponential decay with a factor of 0.75. For each click model, the gap between its examination and click curves in Fig. 3 correspond to the average document relevance for each position. So we expect that top-ranked search results in navigational queries are generally more relevant than informational queries, which leads to higher search relevance scores. And in fact users usually have a higher ex-

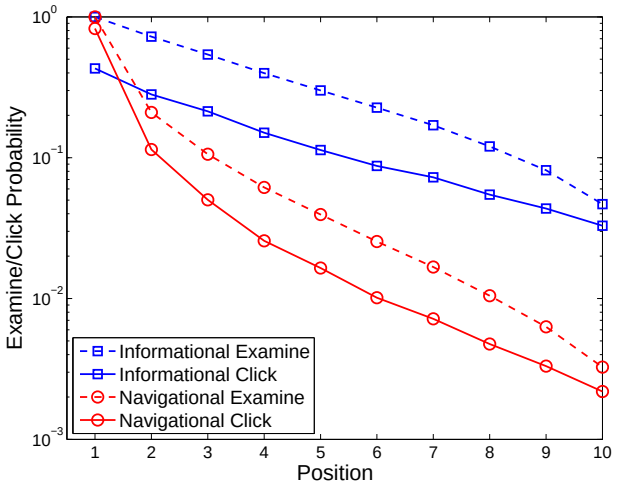

Figure 3: Examination and click probability distributions over the top 10 positions for informational queries and navigational queries.

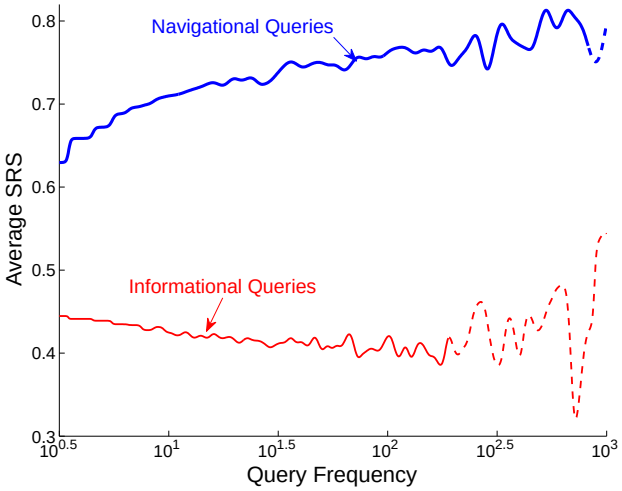

Figure 4: Average Search Relevance Score (SRS) for navigational queries and informational queries.

pectation on the relevance of navigational queries than informational queries. A few additional summary statistics for the two models ("Nav" and "Inf" in short) are as follows:

Summary Statistics	Nav	Inf
Expected First Clicked Position	1.34	2.46
Expected Last Clicked Position	1.47	3.52
Examination Depth	1.48	3.61

We compute search relevance scores using Eq. 9 for each query term and plot how these scores vary with query frequency (the number of corresponding query sessions) in Fig. 4. For each query type, solid lines represent all query terms except the 100 most frequent ones, and dash lines indicate that data points are limited. Both curves are smoothed with Gaussian kernels. As expected, navigational queries generally have higher scores than informational queries. It is interesting to see that trends for two types of queries are in opposite direction. Most popular navigation queries achieve

best quality scores, partly because the search engine might tweak their rankings for some head queries to place the best document on the top. On the other hand, SRS curve for informational queries goes down as query terms are more popular. There is not an easy answer to explain this effect and it might be interesting to investigate further and test some conjectures. For example, there might be potential biases introduced by query resubmission (as suggested by one of our anonymous reviewers) or by the fraction of ignored no-click sessions.

5. CONCLUSION AND DISCUSSION

In this study, we show that by tailoring click models to user goals, better performance can be achieved by fitting different click models for navigational queries and informational queries. Moreover, we provide a quantitative comparison of user behavior using summary statistics derived from click models learned. And we propose the search relevance score to evaluate search engine performance at different granularity levels.

We also tested a number of alternatives in our experiments. For example, we tried to combine the user goal classification of query terms from different metrics by simple voting, however, the MedClk method remains the winner. We also implemented soft query clustering using mixture of Gaussian for MeanClk and AvgClk features, but this did not lead to better log-likelihood. It is also possible to fit completely different click models (*e.g.*, a UBM and a DCM) for different query categories, but we suspect that the results would not outperform the best of the two.

A hidden assumption in the derivation of SRS in Sec. 3 is that document impression, the identity of all documents in the search result, is already known. And examination probabilities are computed *a priori* without knowledge of the click events. This represents the average user behavior under the particular click model. Therefore, if we want to derive SRS for a particular user, a better alternative is to compute the *posterior* probabilities given the actual click sequences. This type of scores could be helpful for personalized search.

We also plan to carry out similar studies on other click models. In DCM, user behavior parameters do not come into relevance estimation, *i.e.*, no λ appears in Eq. 6. But this observation can not be generalized to other click models such as user browsing models. Since document relevance generally depends on user behavior parameters, we expect that tailoring these click models to user goals could also provide more accurate relevance feedbacks. Another direction to go is studying potential applications of SRS. In particular, we could update the score using real-time click logs and plot a SRS curve over time. Anomalies in the average quality may be used to trigger a bug report for further investigation.

6. ACKNOWLEDGMENTS

This study was inspired by an email communication with Nick Craswell. We would like to thank Chao Liu for the discussion that also helped to motivate this work. And we are grateful to anonymous reviewers for the comments on the experimental evaluation as well as the presentation of the paper.

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

[3] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 89–96, 2005.

[4] 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, *NIPS 20*, pages 217–224. 2008.

[5] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *WSDM '08: Proceedings of the 1st international conference on Web search and web data mining*, pages 87–94, 2008.

[6] 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.

[7] F. Guo, C. Liu, and Y.-M. Wang. Efficient multiple-click models in web search. In *WSDM '09: Proceedings of the 2nd international conference on Web search and web data mining*, 2009.

[8] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02*, pages 133–142, 2002.

[9] 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.

[10] 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.

[11] U. Lee, Z. Liu, and J. Cho. Automatic identification of user goals in web search. In *WWW '05*, pages 391–400, 2005.

[12] J. Pitkow and M. Recker. Results from the first world-wide web user survey. In *Selected papers of the first conference on World-Wide Web*, pages 243–254, 1994.

[13] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? In *CIKM '08*, pages 43–52, 2008.

[14] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *WWW '07*, pages 521–530, 2007.

[15] D. E. Rose and D. Levinson. Understanding user goals in web search. In *WWW '04*, pages 13–19, 2004.

[16] 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.