Modeling Clicks Beyond the First Result Page

Aleksandr Chuklin*
Yandex & ISLA, University of Amsterdam
Moscow, Russia
a.chuklin@uva.nl

Pavel Serdyukov
Yandex
Moscow, Russia
pavser@yandex-team.ru

Maarten de Rijke
ISLA, University of Amsterdam
Amsterdam, The Netherlands
derijke@uva.nl

ABSTRACT

Most modern web search engines yield a list of documents of a fixed length (usually 10) in response to a user query. The next ten search results are usually available in one click. These documents either replace the current result page or are appended to the end. Hence, in order to examine more documents than the first 10 the user needs to explicitly express her intention. Although clickthrough numbers are lower for documents on the second and later result pages, they still represent a noticeable amount of traffic.

We propose a modification of the Dynamic Bayesian Network (DBN) click model by explicitly including into the model the probability of transition between result pages. We show that our new click model can significantly better capture user behavior on the second and later result pages while giving the same performance on the first result page.

Categories and Subject Descriptors

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

Keywords

Click models, evaluation, user behavior

1. INTRODUCTION

A result page of a modern search engine (whether commercial or not) usually has buttons leading to more results; Figure 1 shows an example of such buttons. There, the user can switch to the next result page either by using the page number (e.g., "**2**") or by clicking the "**Next**" button. We have a similar setup in our experiments.

By analyzing the click log of the Yandex search engine we learned that one third of all users uses the pagination buttons at least once a week. At the query level, with probability 5–10%, a user will go to the second result page. This number is even bigger for further result pages—once she has switched to the second page, a user often continues to the third and fourth pages, and this probability is at least five times bigger than the probability of switching from the first to the second page.[1] On average, our users examine 1.1 pages. These facts suggest that we need to pay more attention to the ranking of documents below the first result page—such documents have a non-trivial click pattern and are examined by a substantial number of users.

Figure 1: Result page switching (pagination) buttons.

The main research problem of this paper is modeling user behavior on search engine result pages. We argue that existing click models cannot properly model the click skewness caused by the need for switching pages and propose a modification of the Dynamic Bayesian Network click model by explicitly including into the model the probability of transitioning between result pages.

2. RELATED WORK

Seo et al. [8] studied a problem that is similar to the skewness problem we are addressing in the context of aggregated search interfaces. In their setup, the search engine returns a set of *vertical* blocks, each having a so-called *rank cut*—only the top five documents from the vertical are shown. They showed that users rarely continue to a vertical-specific result page, switching to another vertical instead. Hence, they observe skewness in the amount of clicks received by a particular vertical. To deal with this skewness the authors propose to use regularization of the click counts.

Instead, we propose to use a click model approach. A *click model* is a probabilistic model that models user click behavior. Usually these models incorporate observed document click events (C_k) as well as hidden events, e.g., "the user examined the snippet of the k-th document" (E_k). In this paper we use the Dynamic Bayesian Network (DBN) model, due to Chapelle and Zhang [1]. Since its introduction, DBN has been widely used and extended to incorporate different ideas. For example, Hu et al. [6] use the DBN model to show how relevance perception differs accross sessions. More recently, Huang et al. [7] also build a model on top of the DBN model, but with a different motivation from ours; they in-

*Now at Google Switzerland.

[1] Due to the sensitive nature of this information we cannot disclose the exact numbers.

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.
ACM 978-1-4503-2263-8/13/10.
http://dx.doi.org/10.1145/2505515.2507859.

corporate mouse movement information into a click model and show that it leads to better click prediction.

3. METHOD

Background. For each query q we define a query session to be a set of actions performed by the user starting when she issued a query and continuing until she issues another query or abandons the search engine completely (e.g., by closing the browser window). The DBN model [1] considers the following binary events for every ranking position k:

- E_k — the user examines the k-th URL (result snippet)
- A_k — the user is attracted by the k-th URL
- C_k — the user clicks the k-th URL (observed variable)
- S_k — the user is satisfied by the k-th URL

We use the Simplified DBN model described in [1, Section 5] and refer to it as SDBN. This model assumes that the user examines (E_k) documents one by one starting from the first document ($E_1 = 1$). She first examines the document snippet and if she has been attracted by the document ($A_k = 1$), it receives a click ($C_k = 1$). If, after being clicked, the result satisfies the user ($S_k = 1$), the user stops examining documents ($E_j = 0$ for $j > k$). Otherwise she proceeds to the next result. Formally,

$$P(E_1 = 1) = 1 \tag{1}$$

$$P(A_k = 1) = a_q(u_k) \tag{2}$$

$$C_k = 1 \Leftrightarrow A_k = 1 \text{ and } E_k = 1 \tag{3}$$

$$P(S_k = 1 \mid C_k = 0) = 0 \tag{4}$$

$$P(S_k = 1 \mid C_k = 1) = s_q(u_k) \tag{5}$$

$$E_{k+1} = 1 \Leftrightarrow E_k = 1 \text{ and } S_k = 0, \tag{6}$$

where $a_q(u_k)$ is the attractiveness of the k-th document snippet and $s_q(u_k)$ is the satisfaction probability for the k-th document. The a and s parameters are learned from the clicks using Algorithm 1 in [1]. By learning these parameters we can then estimate the document relevance:

$$r_q(u_k) := P(S_k = 1 \mid E_k = 1) = a_q(u_k)s_q(u_k). \tag{7}$$

As was shown in [1, Section 4.3] this predicted relevance can be combined with a standard editorial-based offline learning to rank approach to get better retrieval performance (a 2% gain in DCG score is reported). This combination with click model-based prediction becomes especially important for low ranks—when using depth-K pooling [10] low-rank results may never be judged by human editors.

Beyond the first result page. The most straightforward way of extending a click model beyond the first 10 pages is to simply apply the model to the entire list of documents, not just the first 10. This model implicitly assumes that the user always clicks on the pagination button when she is not satisfied/attracted by every next 10th document. Below, we use it as our *baseline* method.

The first model we propose is an extension of the SDBN model. Essentially, we introduce a fake "document" p corresponding to a pagination button. This fake document is shared across all the result pages, i.e., model parameters for this document depend only on the user query q. Because this is not a real document we assign a different meaning to the model parameters: the attractiveness $a_q(p)$ of the fake document p models the probability of not stopping after examining 10 documents and proceeding to the next page of results by clicking the pagination button (continuation probability); the satisfaction probability $s_q(p)$ plays the role of an "impatience parameter." The reason to add this "impatience parameter" is that the first document on the second page (u_{11} in our case), once shown, is clicked much less than the first document u_1 on the first page (at least 3 times less on our data). This might be motivated not only by the lower quality of the document, but also by a decrease in trust in the documents after the user has had to click on the pagination button: the user abandons search results right after switching to the next result page. This fact also suggests that we cannot simply apply the SDBN model separately to each result page. The resulting model, that we will refer to as SDBN(P), is visualized in Figure 2. To simplify the image, we assume that a result page has only 3 documents, although in reality it has 10.

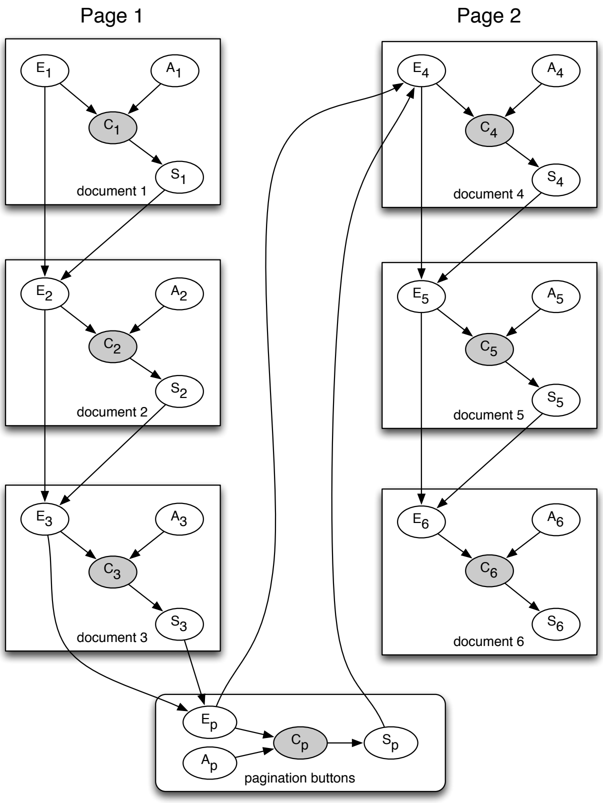

Figure 2: A graphical representation of SDBN(P) and SDBN(P-Q); arrows show probabilistic relations. The probabilities of the events A_p and S_p depend on the query in SDBN(P) and do not depend on the query in SDBN(P-Q).

So, the SDBN(P) model uses the same equations (1)–(6) as SDBN. The click probability for the fake "document" representing the pagination button is also observed: we have not only changed the model, but also introduced an additional source of information. As in [7], where the authors use information about mouse movements to improve the model, we do not measure how well we predict these additional observed events (pagination clicks)—we are only interested in predicting document clicks and do not include pagination clicks in the perplexity computation (10).

We also introduce a second model, SDBN(P-Q), which uses equations (1)–(6) for normal documents and (3), (4), (6), (8), (9) for the fake "document" p representing the pagination button:

$$P(A_p = 1) = a(u_p) \tag{8}$$
$$P(S_k = 1 \mid C_k = 1) = s(u_p), \tag{9}$$

i.e., we assume that the attractiveness and "satisfaction" parameters do not depend on the query (hence, "minus Q" in our notation SDBN(P-Q)) and are global constants.

4. EXPERIMENTS

In this section we compare our SDBN(P) and SDBN(P-Q) models to the original SDBN model. To train the model we use Algorithm 1 from [1]. We then apply the model and see how the adjustments due to pagination buttons correct the click probabilities of the documents below the top-10. The source code of our implementation is available at `https://github.com/varepsilon/clickmodels`.

To compare click models we adopt the commonly used perplexity measure. For each document position k we put:

$$p_k = 2^{-\frac{1}{N}\sum_{j=1}^{N}\left(C_k^j \log_2 q_k^j + (1-C_k^j)\log_2(1-q_k^j)\right)}, \tag{10}$$

where C_k^j is a binary value corresponding to an observed click on the k-th position[2] in the j-th query session, q_k^j is the predicted probability of a click on the k-th position in the j-th query session given the observed previous clicks (computed using Eqs. 1 through 6 for SDBN or corresponding equations for other models). The lower the perplexity of the model, the better. The perplexity of the ideal prediction is 1, while the perplexity of the random coin flip model is 2.

We tested the click models by looking at how well they can predict future events. We recorded the query sessions performed by a small percent of the users of the Yandex search engine during the first 11 days of February, 2013. We had 37,163,170 query sessions resulting in 3,378,470 sessions per day on average. We trained our model using day k and tested it using the next day $k + 1$. The average perplexity values across ten train-test pairs are reported in Figure 3. Vertical bars correspond to the confidence intervals computed using the bootstrap method [5] with 1000 samples and 95% confidence level. We can see that the perplexity values of the models do not differ for positions 1 through 10. However, we do see that the perplexity is much lower for SDBN(P) and SDBN(P-Q) for all positions below rank 10.

Another interesting observation is that we no longer observe constantly decreasing perplexity for $k > 10$. It may well be the case that the underlying cascade hypothesis ("the user examines the documents one by one") needs to be revised for the second and later result pages. We leave this matter as future work.

Next, we contrast the perplexity gains of our two new models over the baseline. If we compute the average perplexity value for positions 1 through 40, then the SDBN(P-Q) and SDBN(P) models yield a 20.5% and 20.2% gain over the baseline, respectively.[3]

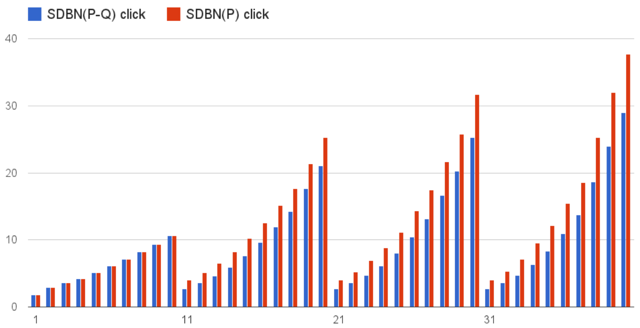

Figure 4: Perplexity of predicting click events for SDBN(P) and SDBN(P-Q) (y axis); document rank (x axis).

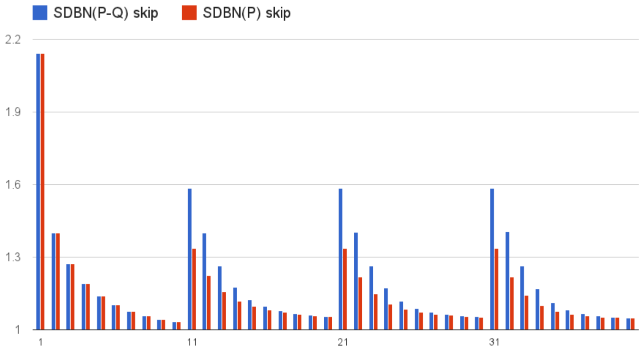

Figure 5: Perplexity of predicting non-click events for SDBN(P) and SDBN(P-Q) (y axis); document rank (x axis).

Following Dupret and Piwowarski [4] we also analyze the prediction perplexity of clicks and skips (non-clicks) separately. We compute p_k^{click} and p_k^{skip} using Eq. 10 limiting ourself to C_k^j equal to 1 and 0 respectively. The corresponding results are reported in Figures 4 and 5. Both models have difficulties predicting clicks, but since click events occur less often than skips, the overall perplexity is not very high. From these plots we can see that SDBN(P) is always better at predicting skips and worse at predicting clicks.

Returning to Figure 3, we can see that on every result page for ranks 20–40, SDBN(P) outperforms SDBN(P-Q) at the beginning of the result page, while SDBN(P-Q) wins at the end of the result page. We believe that the reason is that the relative importance of click/skip prediction is different at the top and the bottom of a result page.[4]

5. DISCUSSION AND CONCLUSION

We studied the skewness of clicks on search result pages. We showed that the click pattern for documents returned below rank 10 is different from the first 10 results usually returned by a web search system. We introduced new click models on top of the widely used DBN model and showed that by explicitly adding pagination buttons into our model we can achieve better results in predicting clicks beyond the first result page. As an immediate application of the click models we can follow the procedure outlined in [3] to build a more accurate evaluation metric.

[2] Here we use continuous numbering, so for the first document of the second page $k = 11$.

[3] To compare the perplexity gain of model B over model A the following value is usually computed: $(p^A - p^B)(p^A - 1)^{-1}$. See, for example, [2, 7].

[4] Recall that SDBN(P-Q) is better at predicting clicks while SDBN(P) predicts non-clicks better.

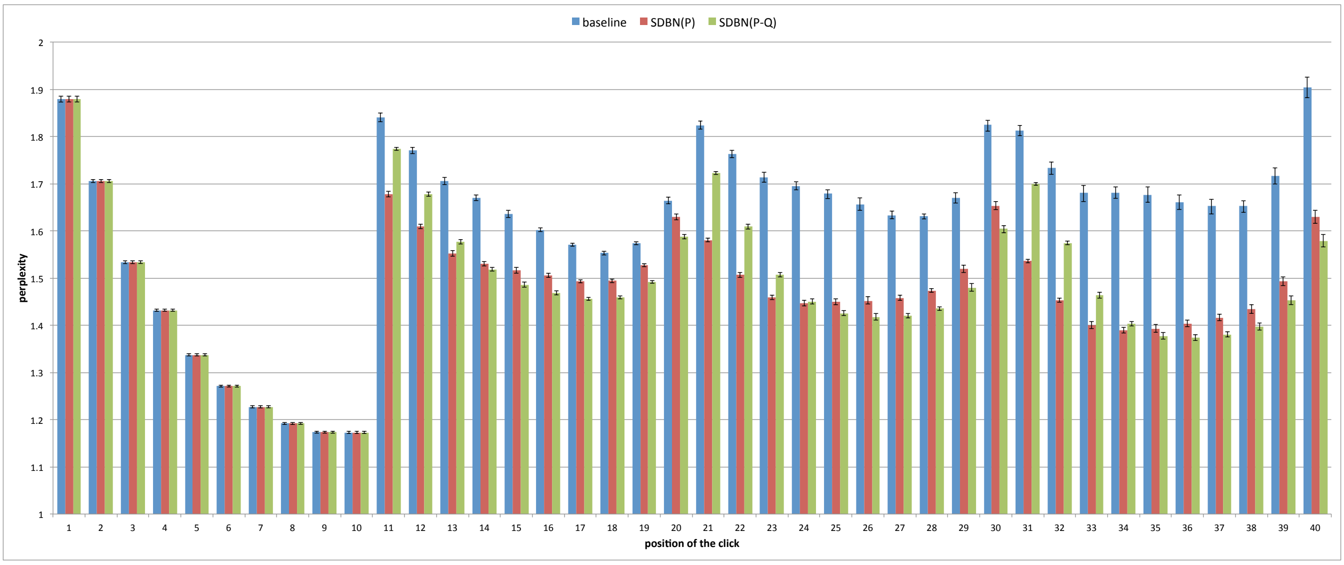

Figure 3: Perplexity of click prediction for different models for positions 1 through 40.

The quality of click prediction is still relatively low compared to the first page, so there is room for improvement. One might want to distinguish between the "**Next**" button and buttons with a page number (see Figure 1). Our internal log studies showed that less than 10% of the users switching to the next page used both the "**Next**" button and page numbers during the one week period covered by our log sample. Most of the users use page numbers (> 70%), less than one third use the "**Next**" button. One could model these two events separately in order to improve performance.

One limitation of our work is that we are focused on a cascade-like DBN model. While the ranks of user clicks generally appear in ascending order [1], out-of-order clicks are more prominently visible when analyzing pagination buttons —users often skip the next result page or return to the previously viewed page (> 10% of the users do it at least once a week, which forces us to ignore 13% of query sessions). It might be interesting to analyze such sessions in detail.

One direction for future work is to consider mobile and/or tablet search. The mobile/tablet version of a search result page usually has a different design from the desktop version of the result page so as to facilitate user input. In particular, the pagination button may look differently: sometimes mobile pagination buttons append new result items to the end of the same result page instead of switching to a new page. It would be interesting to check whether our models are applicable for mobile and tablet search interfaces.

Another future direction is an analysis of different query classes and different users. By having some prior information about the query and the user we can further refine the model by adjusting the probabilities related to the pagination buttons. We can go even further and make all model parameters depend on the user.

Another interesting direction for future work is to link pagination button usage with search engine switching [9]. Use cases for the pagination button may be similar to the cases when users switch to another search provider (dissatisfaction with the results, need for better coverage, etc.).

Acknowledgements. This research was supported by the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreements nr 258191 (PROMISE Network of Excellence) and 288024 (LiMoSINe project), the Netherlands Organisation for Scientific Research (NWO) under project nrs 640.004.802, 727.011.005, 612.001.116, HOR-11-10, the Center for Creation, Content and Technology (CCCT), the QuaMerdes project funded by the CLARIN-nl program, the TROVe project funded by the CLARIAH program, the Dutch national program COMMIT, the ESF Research Network Program ELIAS, the Elite Network Shifts project funded by the Royal Dutch Academy of Sciences (KNAW), the Netherlands eScience Center under project number 027.012.105 and the Yahoo! Faculty Research and Engagement Program.

REFERENCES

[1] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *WWW*. ACM, 2009.

[2] A. Chuklin, P. Serdyukov, and M. de Rijke. Using intent information to model user behavior in diversified search. In *ECIR*, 2013.

[3] A. Chuklin, P. Serdyukov, and M. de Rijke. Click model-based information retrieval metrics. In *SIGIR*. ACM, 2013.

[4] G. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. In *SIGIR*. ACM, 2008.

[5] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, 1 edition, 1994.

[6] B. Hu, Y. Zhang, W. Chen, G. Wang, and Q. Yang. Characterizing search intent diversity into click models. In *WWW*, page 17. ACM, 2011.

[7] J. Huang, R. W. White, G. Buscher, and K. Wang. Improving searcher models using mouse cursor activity. In *SIGIR*, page 195. ACM, 2012.

[8] J. Seo, W. B. Croft, K. H. Kim, and J. H. Lee. Smoothing click counts for aggregated vertical search. *Advances in Information Retrieval*, 2011.

[9] P. Serdyukov, G. Dupret, and N. Craswell. WSCD2013: workshop on web search click data 2013. In *WSDM*. ACM, 2013.

[10] K. Spark Jones and C. van Rijsbergen. Information retrieval test collection. *J. documentation*, 32(1): 59–75, 1976.