Evaluating and Analyzing Click Simulation in Web Search

Stepan Malkevich
St. Petersburg State University
St. Petersburg, Russia
stepamalkevich@yandex.ru

Ilya Markov
University of Amsterdam
Amsterdam, The Netherlands
i.markov@uva.nl

Elena Michailova
St. Petersburg State University
St. Petersburg, Russia
e.mikhaylova@spbu.ru

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

ABSTRACT

We evaluate and analyze the quality of click models with respect to their ability to simulate users' click behavior. To this end, we propose distribution-based metrics for measuring the quality of click simulation in addition to metrics that directly compare simulated and real clicks. We perform a comparison of widely-used click models in terms of the quality of click simulation and analyze this quality for queries with different frequencies. We find that click models fail to accurately simulate user clicks, especially when simulating sessions with no clicks and sessions with a click on the first position. We also find that click models with higher click prediction performance simulate clicks better than other models.

CCS CONCEPTS

•Information systems →Web search engines; Information retrieval; Retrieval models and ranking;

1 INTRODUCTION

Simulation has long played a role in information retrieval (IR). Queries have been simulated [2] and so have labeled test collection [3]. Several recent workshops and tutorials have focused on simulating user interactions and search behavior; see, e.g., [1]. Simulating user search behavior in information retrieval and web search is crucial not only for academic researchers, who may not have access to large-scale search logs, but also for commercial systems, where online experiments with real users cannot scale infinitely. Clicks are a particularly important aspect of online user behavior; clicks are widely used both to evaluate the quality of search [13], as a predictive signal [8], and to improve the quality of search [15].

To simulate clicks, researchers and practitioners use clicks models [6, 16]. For example, such simulated clicks have been recently used in interleaving experiments [7, 14]. Little attention has been given to the problem of click simulation on its own and to measuring the quality of this simulation [16]. It is not clear, for instance, which click models simulate clicks better and why, i.e., what theoretical properties of click models affect the quality of click simulation. In this paper we fill this gap by evaluating and analyzing the simulation quality of the most widely-used and well-performing click models, namely, DBN [5], CCM [12] and UBM [10].

Our contributions are the following. First, we propose to consider distribution-based metrics to measure the quality of click simulation in addition to metrics that directly compare simulated and real clicks. Second, we perform a thorough comparison of the abovementioned click models in terms of the quality of click simulation.

2 RELATED WORK

Click simulation is used to either evaluate the quality of click models [16, 17] or to simulate online experiments in cases where real users are not available [7, 14]. Zhu et al. [17] simulated clicks to estimate the query-document CTR and then compared this estimated CTR to the real CTR value. Xing et al. [16] directly evaluated click simulation by calculating the mean average error (MAE) between the ranks of the first/last simulated clicks and the ranks of the real clicks. In this work we take a close look at click simulation and follow the latter approach to evaluate the simulation quality. In addition, we propose distribution-based measures for evaluation of click simulation.

Hofmann et al. [14] proposed probabilistic interleaving and used simulated clicks to evaluate the proposed method. Similarly, Chuklin et al. [7] simulated clicks to evaluate vertical-aware interleaving. However, the quality of click simulation itself was not taken into account. In this work, we aim to directly evaluate this quality, which will further inform and support various applications of click simulation.

3 BACKGROUND

Click models. We consider click models that have been shown to have the best performance in terms of modeling and predicting clicks [6]: the user browsing model (UBM) [10], the dynamic Bayesian network model (DBN) [5], and the click chain model (CCM) [12].

The UBM model includes two types of parameters: (i) the probability $\gamma_{rr'}$ of examining a snippet of a search result, which depends on the rank r of the result and on the rank r' of the previously clicked result, and (ii) the probability α_{qd} of snippet d being attractive to a user given query q. According to UBM, a snippet is clicked if, and only if, it is both examined and attractive, i.e., with

ICTIR'17, October 1–4, 2017, Amsterdam, Netherlands.
 ISBN 978-1-4503-4490-6/17/10…$15.00
DOI: https://doi.org/10.1145/3121050.3121096

probability $\gamma_{rr'}\alpha_{qd}$. UBM was reported to outperform DBN and CCM in terms of log-likelihood and perplexity [6, 11].

DBN considers not only the attractiveness probability of a snippet α_{qd}, but also the satisfactoriness probability σ_{qd} of the actual search result d after it is clicked. DBN follows the cascade assumption [9] and assumes that a user examines a snippet at rank r with probability γ if, and only if, she examined a snippet at rank $r-1$ and was not satisfied with it (this happens with probability $1-\alpha_{qd}\sigma_{qd}$). Similarly, according to UBM, a snippet is clicked if, and only if, it is both examined and attractive, where the examination probability is calculated recursively.

The CCM model also contains the set of attractiveness parameters α_{qd} and, similarly to DBN, follows the cascade assumption. CCM introduces three examination parameters $\tau_1-\tau_3$ that determine the examination probability for a snippet at rank r based on the examination and attractiveness of the snippet at rank $r-1$. Just as before, a snippet is clicked if, and only if, it is both examined and attractive.

Click simulation. We simulate clicks using Algorithm 1, as described in [6]. For each rank r (line 1) the algorithm calculates

Algorithm 1 Simulating user clicks for a query session.

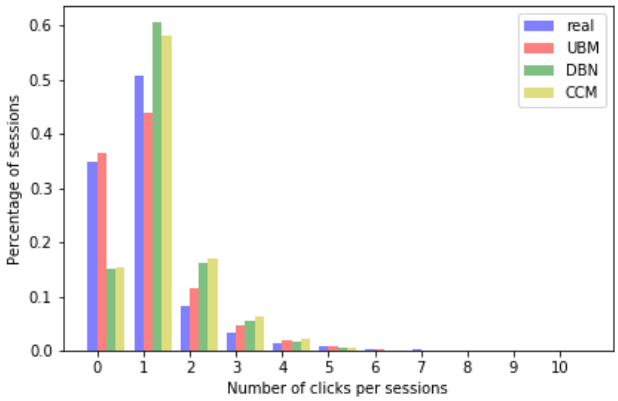
Input: click model M, query session s
Output: vector of simulated clicks $(c_1, \ldots, c_n)$
1: **for** $r \leftarrow 1$ *to* $|s|$ **do**
2: Compute $p = P(C_r = 1 \mid C_1 = c_1, \ldots, C_{r-1} = c_{r-1})$ using previous clicks $c_1, \ldots, c_{r-1}$ and the parameters of model M
3: Generate random value c_r from $Bernoulli(p)$

the probability p of clicking on that rank, given all previous clicks $c_1, \ldots, c_{r-1}$ (line 2). This probability is calculated using a click model M; all click models are able to calculate the conditional probability of a click on a result using previously observed clicks in the same query session. Based on the obtained conditional click probability p the algorithm generates a random value from a Bernoulli distribution with parameter p (line 3). This random value indicates the presence of a click. If it equals 1, then model M simulates a click, otherwise model M simulates the absence of a click.

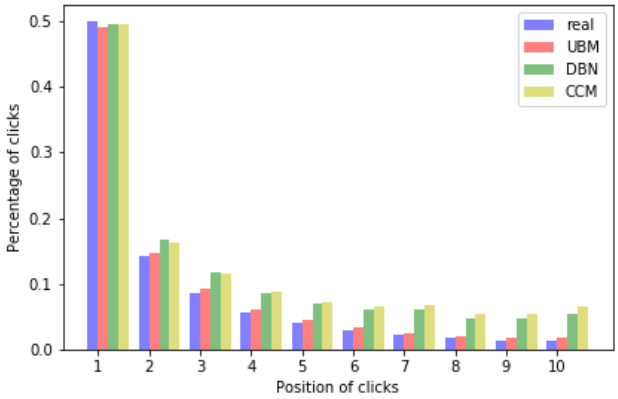
4 EVALUATION METRICS

The quality of click simulation is usually measured by comparing real clicks and simulated clicks. In particular, Xing et al. [16] measure the mean absolute error (MAE) in predicting the first clicked rank and the last clicked rank. We consider these metrics in our work. Note that for sessions without clicks we assume that the rank of the first/last click is zero.

At the same time, we argue that directly comparing real and simulated clicks is too strict and a good click simulator does not necessarily have to have (and even should not have) low MAE of the first/last clicked rank. Consider that we observe two sessions for the same query. In the first session we observe a click at some rank, while in the second session we do not observe any click. Assume that a click simulator simulates an opposite situation: no clicks in the first session and a click at the same rank in the second session. In this case, MAE will be large, because, strictly speaking, the clicks



(a) Distributions of clicks over sessions.



(b) Distribution of clicks over ranks.

Figure 1: Click distributions considered in this paper.

are not simulated correctly. On the other hand, the distribution of clicks among sessions and over ranks is preserved. Moreover, we argue that a realistic click simulator must preserve the distribution of clicks and should not necessarily simulate exactly same clicks in exactly same sessions. Therefore, we propose to measure the Kullback-Leibler divergence between the distribution of real clicks and the distribution of simulated clicks in addition to measuring MAE of the first/last clicked ranks.

There are two natural click distributions to look at. First, the distribution of clicks over sessions (see Figure 1(a)), which shows the percentage of sessions with a certain number of clicks. Second, the distribution of clicks over ranks (see Figure 1(b)), which shows how many times a certain rank was clicked.

Note that directly measuring the KL-divergence between the real and simulated distribution of clicks does not give any meaningful information. Consider the following example. The actual clicks for a query q_1 are $[1, 0, 1, 0, 0, 0, 0, 0, 0, 0]$, while a simulator predicts $[0, 1, 0, 1, 0, 0, 0, 0, 0, 0]$. For a query q_2 the situation is reversed: the actual clicks are $[0, 1, 0, 1, 0, 0, 0, 0, 0, 0]$, while the predicted clicks are $[1, 0, 1, 0, 0, 0, 0, 0, 0, 0]$. In this case, the global KL-divergence between the distribution of real clicks and the distribution of simulated clicks is 0.0, while the simulation is actually wrong. For this reason, we measure a local KL-divergence for every query and then

calculate a weighted average of local divergences as follows:

$$KL\text{–}div = \frac{\sum_{q \in Q} KL\text{–}div(q) \cdot s_q}{\sum_{q \in Q} s_q} \quad (1)$$

where Q is the number of unique queries and s_q is the number of sessions observed for a particular query q. We calculate this metric for both click distributions: distribution over sessions and distribution over ranks. Lower values of the metrics denote better click simulation performance.

5 EXPERIMENTAL SETUP

Dataset. In our experiments, we use a publicly available click log published by Yandex[1] within the personalized web search challenge.[2] This click log consists of 27 days of search activity, where each day contains more than 2 million query sessions (2,413,800 on average). We split the data in two parts: first 14 days for training and last 13 days for testing. We train click models on all query sessions of the first 14 days (33,310,079 sessions). Due to limited computational resources we measure the quality of click simulation on the first 100K query sessions of each test day, which results in 1,300,000 sessions for testing. Since query sessions are independent and their order within a day is not specified, using the first 100K sessions of a day is equivalent to considering random sessions.

Baselines. We compare the simulation performance of the DBN, CCM and UBM models to the following naïve baselines: (i) always simulate no clicks (**baseline 1**), and (ii) always simulate a click on the first position (**baseline 2**). We considered these simple simulators as baselines due to specific properties of the data: 35% of the test sessions have no clicks and 51% of the test sessions have a click on the first position.

6 EXPERIMENTAL RESULTS AND ANALYSIS

In this section we first compare the click simulation performance of the UBM, DBN and CCM click models to that of the baselines presented above. Then we discuss the relative performance of click models compared to each other.

Baselines vs. click models. Table 1 summarizes the simulation performance of the click models and baselines in terms of the MAE of the first/last clicked rank and the KL-divergence of the click distribution over sessions (session-based KL) and over ranks (rank-based KL).

Table 1 shows that Baseline 2, which always simulates a click on the first position, achieves the best performance in terms of all metrics. This is not surprising as 51% of the sessions in the test set contain a click on the first position. An interesting result here is that advanced click models, such as UBM, DBN and CCM, cannot simulate sessions with a click on the first position as accurate as Baseline 2.

Baseline 1, which simulates no clicks, performs worse than Baseline 2, because there are fewer sessions without clicks (35%) than with a click on the first position (51%). Baseline 1 also outperforms the click models in terms of all but one metric, which means that the considered click models cannot accurately simulate sessions without clicks either. (The rank-based KL-divergence of Baseline 1 is worse than that of the click models, because this baseline does not simulate any click at any rank).

Table 1: Simulation performance of click models.

	MAE		KL-divergence	
Model	First rank	Last rank	Session-based	Rank-based
Baseline 1	0.97	1.36	1.79	1.63
Baseline 2	0.69	1.18	1.61	0.85
UBM	0.96	1.69	1.84	1.26
DBN	1.55	2.34	1.94	1.44
CCM	1.54	2.42	1.99	1.43

In Tables 2 and 3 we take a closer look at the click simulation quality for queries with different frequencies: frequent queries with more than 100 sessions, torso queries with 10–100 sessions and rare queries with 1–10 sessions. Here the situation is slightly different. Baseline 2 is still the best in almost all cases, apart from the session-based KL-divergence for frequent queries (Table 3). Baseline 1 is better than click models in terms of MAE, but not in terms of KL-divergence for frequent and torso queries.

Overall, naïve baselines almost always outperform advanced click models in terms of both MAE and KL-divergence apart from some cases for frequent and torso queries. These results suggest the following. First, click models cannot accurately simulate sessions with no clicks and with a click on the first position, which is crucial in a web search scenario, where such sessions comprise the majority of cases. Ways of improving the simulation accuracy of click models in these cases should be considered. Second, the metrics used in this study do not take into account the "usefulness" of simulated clicks. Naïve baselines that always simulate the same click pattern are not useful in practical applications, such as training a ranker using clicks.

Table 2: MAE of the first/last clicked rank for queries with different frequencies.

	Frequent (100+)		Torso (10–100)		Rare (1–10)	
Model	First	Last	First	Last	First	Last
Baseline 1	0.93	1.12	0.97	1.28	0.99	1.51
Baseline 2	0.52	0.76	0.68	1.07	0.78	1.45
UBM	0.72	1.12	0.92	1.53	1.12	2.06
DBN	1.07	1.55	1.44	2.10	1.85	2.89
CCM	1.07	1.58	1.44	2.16	1.84	3.00

Table 3: Session- and rank-based KL-divergence for queries with different frequencies.

	Frequent (100+)		Torso (10–100)		Rare (1–10)	
Model	Session	Rank	Session	Rank	Session	Rank
Baseline 1	0.0046	0.0211	0.08	0.35	3.67	3.18
Baseline 2	0.0021	0.0017	0.05	0.05	3.33	1.74
CCM	0.0008	0.0048	0.07	0.15	4.12	2.88
DBN	0.0007	0.0046	0.06	0.15	4.02	2.91
UBM	0.0007	0.0019	0.06	0.11	3.80	2.55

[1] http://yandex.com
[2] https://www.kaggle.com/c/yandex-personalized-web-search-challenge

Relative performance of click models. First, it has to be noted that the mean absolute errors for the UBM model, presented in Table 1, are higher than those reported in [16]. We believe this is because we use a different dataset compared to [16].

Second, it is clear that UBM simulates clicks better than DBN and CCM in terms of both MAE and KL-divergence. This also holds for all query frequencies (see Tables 2 and 3). UBM was reported to outperform DBN and CCM in terms of click prediction performance, measured by log-likelihood and perplexity [6, 11]. Although there is an intuitive relation between the quality of click prediction and the quality of click simulation, this intuition has not been confirmed so far. Our results show that indeed the best-performing click model in terms of log-likelihood and perplexity is also the best-performing click simulator in terms of MAE and KL-divergence.

Third, Tables 1–3 show that DBN and CCM perform similarly to each other in terms of click simulation. They were also shown to have similar log-likelihood and perplexity when predicting clicks [6, 11]. This further confirms the above discussion.

Finally, Tables 2 and 3 show that click models simulate clicks best for frequent queries and worst for rare queries. This is a known limitation of click models: they do not perform well for rare queries [11]. Our results confirm that this also holds for click simulation.

Overall, our results confirm the intuition that a click model with better log-likelihood and perplexity in a click prediction task should also simulate clicks better.

7 CONCLUSION

We considered the problem of click simulation in web search and studied the ability of click models to simulate clicks. To measure the quality of click simulation, we used metrics that directly compare real and simulated clicks and proposed to use distribution-based metrics that compare the distributions of real and simulated clicks. We compared the simulation quality of the best-performing and widely-used click models, namely UBM, DBN and CCM, and analyzed this performance for queries with different frequencies.

Our main findings are the following. First, click models do not simulate accurately sessions without clicks and sessions with a click on the first position. In a web search scenario, considered in this work, such sessions comprise the majority of oberved sessions. Thus, in this scenario, naïve baselines that simulate no clicks and a click on the first position outperform advanced click models in terms of click simulation. This finding suggests that click models should consider ways of dealing with sessions with no clicks and sessions with a click on the first position. Second, we confirmed the intuition that click models that have high click prediction performance (usually measured with log-likelihood and perplexity) also have the best click simulation performance. Particularly, the UBM model was shown to be the best in predicting clicks and it appears to be the best in simulating clicks compared to DBN and CCM. This finding suggests that building click models with high log-likelihood will result in better click simulators.

There are a few directions for future work. First, we plan to study click simulation in scenarios other than web search, e.g., exploratory search, academic search, search in digital libraries and archives. We expect that user click behavior in these scenarios will differ significantly from that in web search. In particular, there may be much fewer sessions with no clicks or only one click. Second, in addition to directly evaluating the performance of click simulators we plan to consider and evaluate various applications of click simulation, e.g., building rankers using simulated clicks. This way we will evaluate the "usefulness" of simulated clicks. Finally, we plan to consider recently proposed neural click models [4] for the task of click simulation and evaluate their simulation performance. We expect that these neural models will capture more complex patterns in user click behavior and, therefore, simulate clicks better. In particular, we expect them to better deal with sessions with no clicks or only one click.

Acknowledgments. This research was supported by Ahold Delhaize, Amsterdam Data Science, the Bloomberg Research Grant program, the Criteo Faculty Research Award program, the Dutch national program COMMIT, Elsevier, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 312827 (VOX-Pol), the Microsoft Research Ph.D. program, the Netherlands Institute for Sound and Vision, the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.-001.116, HOR-11-10, CI-14-25, 652.002.001, 612.001.551, 652.001.003, and Yandex. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Leif Azzopardi. 2016. Simulation of Interaction: A Tutorial on Modelling and Simulating User Interaction and Search Behaviour. In *SIGIR*. ACM, 1227–1230.
[2] Leif Azzopardi, Maarten de Rijke, and Krisztian Balog. 2007. Building simulated queries for known-item topics: An analysis using six European languages. In *SIGIR*. ACM.
[3] Richard Berendsen, Manos Tsagkias, Wouter Weerkamp, and Maarten de Rijke. 2013. Pseudo test collections for training and tuning microblog rankers. In *SIGIR*. ACM, 53–62.
[4] Alexey Borisov, Ilya Markov, Maarten de Rijke, and Pavel Serdyukov. 2016. A Neural Click Model for Web Search. In *WWW*. 531–541.
[5] Olivier Chapelle and Ya Zhang. 2009. A dynamic bayesian network click model for web search ranking. In *WWW*. 1–10.
[6] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. *Click Models for Web Search*. Morgan & Claypool.
[7] Aleksandr Chuklin, Anne Schuth, Katja Hofmann, Pavel Serdyukov, and Maarten de Rijke. 2013. Evaluating Aggregated Search Using Interleaving. In *CIKM*. ACM Press, New York, NY, USA.
[8] Aleksandr Chuklin, Pavel Serdyukov, and Maarten de Rijke. 2013. Click model-based information retrieval metrics. In *SIGIR*. 493–502.
[9] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *WSDM*. 87–94.
[10] Georges E. Dupret and Benjamin Piwowarski. 2008. A user browsing model to predict search engine click data from past observations. In *SIGIR*. 331–338.
[11] Artem Grotov, Aleksandr Chuklin, Ilya Markov, Luka Stout, Finde Xumara, and Maarten de Rijke. 2015. A Comparative Study of Click Models for Web Search. In *CLEF*. 78–90.
[12] Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Wang, and Christos Faloutsos. 2009. Click chain model in web search. In *WWW*. 11–20.
[13] Katja Hofmann, Lihong Li, and Filip Radlinski. 2016. Online Evaluation for Information Retrieval. *Foundations and Trends in Information Retrieval* 10, 1 (2016), 1–117.
[14] Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. 2011. A probabilistic method for inferring preferences from clicks. In *CIKM*. ACM Press, New York, NY, USA.
[15] Thorsten Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. In *KDD*. 133–142.
[16] Qianli Xing, Yiqun Liu, Jian-Yun Nie, Min Zhang, Shaoping Ma, and Kuo Zhang. 2013. Incorporating User Preferences into Click Models. In *CIKM*. 1301–1310.
[17] Zeyuan Allen Zhu, Weizhu Chen, Tom Minka, Chenguang Zhu, and Zheng Chen. 2010. A Novel Click Model and Its Applications to Online Advertising. In *WSDM*. 321–330.