A Comparative Study of Click Models for Web Search

Artem Grotov, Aleksandr Chuklin, Ilya Markov(✉), Luka Stout, Finde Xumara, and Maarten de Rijke

University of Amsterdam, Amsterdam, The Netherlands
{a.groto,a.chuklin,i.markov,derijke}@uva.nl, lukastout@gmail.com, finde.findexumara@student.uva.nl

Abstract. Click models have become an essential tool for understanding user behavior on a search engine result page, running simulated experiments and predicting relevance. Dozens of click models have been proposed, all aiming to tackle problems stemming from the complexity of user behavior or of contemporary result pages. Many models have been evaluated using proprietary data, hence the results are hard to reproduce. The choice of baseline models is not always motivated and the fairness of such comparisons may be questioned. In this study, we perform a detailed analysis of all major click models for web search ranging from very simplistic to very complex. We employ a publicly available dataset, open-source software and a range of evaluation techniques, which makes our results both representative and reproducible. We also analyze the query space to show what type of queries each model can handle best.

1 Introduction

Modeling user behavior on a search engine result page (SERP) is important for understanding users, supporting simulation experiments [12,11], evaluating web search results [1,4] and improving document ranking [2,7]. In recent years, many models of user clicks in web search have been proposed [3]. However, no comprehensive evaluation of these click models has been performed using publicly available datasets and a common set of metrics with a focus on an analysis of the query space. As a result, it is not clear what the practical advantages and drawbacks are of each proposed model, how different models compare to each other, which model should be used in which settings, etc.

In this paper we aim to compare the performance of different click models using a common dataset, a unified implementation and a common set of evaluation metrics. We consider all major click models for web search ranging from simple the Click-Through Rate model (CTR), Position-Based Model (PBM) and Cascade Model (CM) [5] through the more advanced Dependent Click Model (DCM) [10] to more complex User Browsing Model (UBM) [8], Dynamic Bayesian Network model (DBN) [2], and Click Chain Model (CCM) [9].

A. Chuklin—Currently at Google Switzerland.

J. Mothe et al. (Eds.): CLEF 2015, LNCS 9283, pp. 78–90, 2015.
DOI: 10.1007/978-3-319-24027-5_7

Table 1. Notation used in the paper.

Symbol	Description	Symbol	Description
u	A document	E	A random variable for document examination
q	A query	R	A random variable for document relevance
s	A search query session	C	A random variable for a click on a document
j	A document rank	ϵ	The examination parameter
c	A click on a document	r	The relevance parameter
$\mathcal{S}$	A set of sessions		

These models are evaluated using log-likelihood, perplexity, click-through rate prediction, relevance prediction, ranking performance and computation time.

We also analyze two different factors that influence performance of click models, namely, query frequency and click entropy. Intuitively, it is easier to predict clicks for frequent queries than for less frequent ones because of the larger size of the training data and the relatively more uniform click patterns associated with frequent queries. Click entropy can be used to distinguish between navigational and informational queries. Navigational queries tend to have low click entropy (usually only the top result is clicked), while informational queries tend to have high click entropy (several results may be clicked before a user's information need is satisfied).

Our main finding is that no single model excels on each of the considered metrics and that sometimes simple models outperform complex ones and that the relative performance of models can be influenced by the data set characteristics such as query frequency and click entropy. These results can guide the application of existing click models and inform the development of new click models.

2 Click Models

In this section, we give an overview of all major click models for web search, which we will then use in our comparative study

Click-Through Rate Models. Three simple click models, all based on click-through rates, predict click probabilities by counting the ratio of clicks to the total number of impressions. In the simplest case of Global CTR (GCTR) this ratio is computed globally for all documents, while in Rank CTR (RCTR) it is computed separately for each rank j and in Document CTR (DCTR) for each document-query pair uq:

$$P_{GCTR}(C_u = 1) = r = \frac{1}{\sum_{s \in \mathcal{S}} |s|} \sum_{s \in \mathcal{S}} \sum_{u \in s} c_{uq} \tag{1}$$

$$P_{RCTR}(C_{u_j} = 1) = r_j = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} c_j \tag{2}$$

$$P_{DCTR}(C_u = 1) = r_{uq} = \frac{1}{|\mathcal{S}_{uq}|} \sum_{s \in \mathcal{S}_{uq}} c_{uq}, \text{ where}, \mathcal{S}_{uq} = \{s_q : u \in s_q\} \tag{3}$$

Position-Based Model. This model builds upon the CTR models and unites DCTR with RCTR. It adds a separate notion of examination probability (E)

which is subject to *position bias* where documents with smaller rank are examined more often; the document can only be clicked if it was examined and is relevant:

$$C_{uq} = 1 \Leftrightarrow (E_{j_u} = 1 \text{ and } R_{uq} = 1) \tag{4}$$

The examination probability $\epsilon_j = P(E_{j_u} = 1)$ depends on the rank j, while the relevance $r_{uq} = P(R_{uq} = 1)$ depends on the document-query pair. Inference of this model is done using the Expectation Maximization algorithm (EM).

Cascade Model. The Cascade Model [5, CM] is another extension to the CTR models. The model introduces the *cascade hypothesis*, whereby a user examines a search result page (SERP) from top to bottom, deciding whether to click each result before moving to the next one; users stop examining a SERP after first click. Inference of the parameters of CM is done using Maximum Likelihood Estimation (MLE). The click probability is defined using the examination (4) and the cascade assumptions:

$$P(E_1 = 1) = 1 \tag{5}$$

$$P(E_j = 1 \mid E_{j-1} = e, C_{j-1} = c) = e \cdot (1 - c), \tag{6}$$

where e and c are 0 or 1, and the only parameters of the models are $r_{uq} = P(R_{uq} = 1)$. The fact that users abandon a search session after the first click implies that the model does not provide a complete picture of how multiple clicks arise in a query session and how to estimate document relevance from such data.

User Browsing Model. [8] propose a click model called the User Browsing Model (UBM). The main difference between UBM and other models is that UBM takes into account the distance from the current document u_j to the last clicked document $u_{j'}$ for determining the probability that the user continues browsing:

$$P(E_{j_u} = 1 \mid C_{u_{j'}} = 1, C_{u_{j'+1}} = 0, \ldots, C_{u_{j-1}q} = 0) = \gamma_{jj'}. \tag{7}$$

Dependent Click Model. The Dependent Click Model (DCM) by [10] is an extension of the cascade model that is meant to handle sessions with multiple clicks. This model assumes that after a user clicked a document, they may still continue to examine other documents. In other words, (6) is replaced by

$$P(E_j = 1 \mid E_{j-1} = e, C_{j-1} = c) = e \cdot (1 - c + \lambda_j c), \tag{8}$$

where λ_j is the continuation parameter, which depends on the rank j of a document.

Click Chain Model. [9] further extend the idea of DCM into the Click Chain Model (CCM). The intuition behind CCM is that the chance that a user continues after a click depends on the relevance of the previous document and that a user might abandon the search after a while. This model can be formalized with (4) and the following conditional probabilities:

$$P(E_{j_u+1} = 1 \mid E_{j_u} = 1, C_{uq} = 0) = \tau_1 \tag{9}$$

$$P(E_{j_u+1} = 1 \mid E_{j_u} = 1, C_{uq} = 1) = \tau_2(1 - r_{uq}) + \tau_3 r_{uq}. \tag{10}$$

Dynamic Bayesian Network Model. The Dynamic Bayesian Network model [2] takes a different approach in extending the cascade model. Unlike CCM, DBN assumes that the user's perseverance after a click depends not on the relevance r_{uq}, but on a different parameter s_{uq} called satisfaction parameter. While r is mostly defined by the snippet on the SERP, the satisfaction parameter s depends on the actual document content available after a click. The DBN model is defined by (4) and the following formulas:

$$P(E_{j_u+1} = 1 \mid E_{j_u} = 1, C_{uq} = 0) = \gamma \tag{11}$$

$$P(E_{j_u+1} = 1 \mid E_{j_u} = 1, C_{uq} = 1) = \gamma(1 - s_{uq}), \tag{12}$$

where γ is a continuation probability after a non-satisfactory document (either no click, or click, but no satisfaction).

In general, the inference should be done using the EM algorithm. However, if γ is set to 1, the model allows easy MLE inference. We refer to this special case as the Simplified DBN model (SDBN).

3 Evaluation Measures

Different studies use different metrics to evaluate click models [3]. In this section we give an overview of these metrics. We will then use all of them in our comparative study.

Log-likelihood. Log-likelihood evaluates how well a model approximates observed data. In our case, it shows how well a click model approximates clicks of actual users. Given a model M and a set of observed query sessions $\mathcal{S}$, log-likelihood is defined as follows:

$$\mathcal{LL}(M) = \sum_{s \in \mathcal{S}} \log P_M\left(C_1, \ldots, C_n\right), \tag{13}$$

where P_M is the probability of observing a particular sequence of clicks $C_1, \ldots, C_n$ according to the model M.

Perplexity. Perplexity measures how surprised a model is to see a click at rank r in a session s [8]. It is calculated for every rank individually:

$$p_r(M) = 2^{-\frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \left(c_r^{(s)} \log_2 q_r^{(s)} + \left(1 - c_r^{(s)}\right) \log_2 \left(1 - q_r^{(s)}\right)\right)}, \tag{14}$$

where $c_r^{(s)}$ is the actual click on the document at rank r in the session s, while $q_r^{(s)}$ is the probability of a user clicking the document at rank r in the session s as predicted by the model M, i.e., $q_r^{(s)} = P_M(C_r = 1)$.

The total perplexity of a model is defined as the average of perplexities over all positions. Lower values of perplexity correspond to higher quality of a click model.

Click-trough Rate Prediction. Click-through rate (CTR) is a ratio of the cases when a particular document was clicked to the cases when it was shown. In [2], the following procedure was proposed to measure the quality of click models using CTRs:

- Consider a document u that appears both on the first position and on some other positions (in different query sessions).
- Hold out as a test set all the sessions in which u appears on the first position.
- Train a click model M on the remaining sessions.
- Use the model M to predict clicks on the document u on the held-out test set (predicted CTR).
- Compute the actual CTR of u on the held-out test set.
- Compute the Root-Mean-Square-Error (RMSE) between the predicted and actual CTRs.

Relevance Prediction. It was noticed in [2] that click models can approximate document relevance. A straightforward way to evaluate this aspect is to compare document relevance as predicted by a model to document relevance labels provided by human annotators. We measure the agreement between the two using the Area Under the ROC Curve (AUC) and Pearson correlation.

Predicted Relevance as a Ranking Feature. The predicted relevance can also be used to rank documents [2]. The performance of such a ranker can be evaluated using any standard IR measure, such as MAP, DCG, etc. In this study, we use NDCG@5 [13]. To calculate NDCG@5 we only consider documents for which we have relevance labels. The evaluation is performed as follows:

- Retrieve all sessions that have complete editorial judgments.
- Sort sessions by session id
- The first 75% are training sessions, the remainder are test sessions.
- Train the model on the training sessions and predict relevance for the test sessions.
- Sort the documents w.r.t the predicted relevance given by the model.
- Compute the NDCG@5.
- Average over all sessions.

Computation Time. Historically, in machine learning a big problem in creating accurate models was the amount of data that was available. However, this is no longer the case, and now we are mostly restricted by the time it takes to learn a model based on a large amount of available data. This makes the ability to efficiently compute parameters an important feature of a successful model. Therefore, we also look at the time it takes to train a click model.

4 Experimental Setup

Our goal is to evaluate and compare the click models presented in Section 2 using the evaluation metrics described in Section 3. To this end we use the first 32 million query sessions from the 2011 Yandex Relevance Prediction contest.[1] In this contest participants were asked to predict document relevance based on click log data. We split the session set into 32 batches of one million sessions each and measured, for every click model, the log-likelihood, perplexity, RMSE of

[1] http://imat-relpred.yandex.ru/en/datasets

CTR prediction and computation time for each of the batches. Then we average the measurements across the batches.

The sessions in each batch are sorted based on their session id and divided into a set of training sessions used to train the click models and a set of test sessions used in the evaluation of the models; the number of sessions in these sets have a 3 to 1 ratio.

To measure the quality of relevance prediction and ranking performance we use sessions for which all the documents have relevance labels. For each query all except the last session is used for training and the last session is used for testing. There are 860861 search sessions and 178 unique queries in the training set and 112 queries in the test set.

To determine whether observed differences are statistically significant we use the two-tailed student-t test with p values below 0.05 indicating significant differences. The error bars in the plots below are standard errors of the means.

Performance Impacting Factors. To evaluate the effect of *query frequency* on click model performance, we split the data into four parts (see Table 2).

Another factor that may influence click model performance is *click entropy*. Click entropy has been used to analyze queries in [6]. The formal definition of the entropy of query q is:

$$ClickEntropy(q) = -\sum_{d \in \mathcal{P}(q)} P(d \mid q) \log_2 P(d \mid q) \tag{15}$$

where $\mathcal{P}(q)$ are documents clicked on for query q and $P(d \mid q)$ is the fraction of clicks on document d among all clicks on q, $P(d \mid q) = \sum_p c_{r_d}^{(q)} \cdot (\sum_{u \in \mathcal{P}(q)} c_{r_u}^{(q)})^{-1}$. Click entropy can be used to distinguish navigational and informational queries. In navigational queries users know what they are looking for so the click entropy will be low because almost all clicks within that query will be on the same document. In an informational query the users explore different results to find the optimal one because they do not know what document they are looking for yet. This gives these queries a high click entropy. We divide our search sessions into three bins with respect to click entropy and report on evaluation measures per bin; statistics of these bins are listed in Table 3.

Table 2. The distribution of session with respect to query frequency.

Query frequency	Number of sessions
2	6944438
3–5	12750938
6–19	16592812
20+	108132750

Table 3. The distribution of session with respect to click entropy.

Click entropy	Number of sessions
0–1	53380500
1–2	48844812
2+	42195625

5 Results

In this section we present the results of our experiments. For every evaluation measure we report the influence of the query frequency and click entropy. Table 4

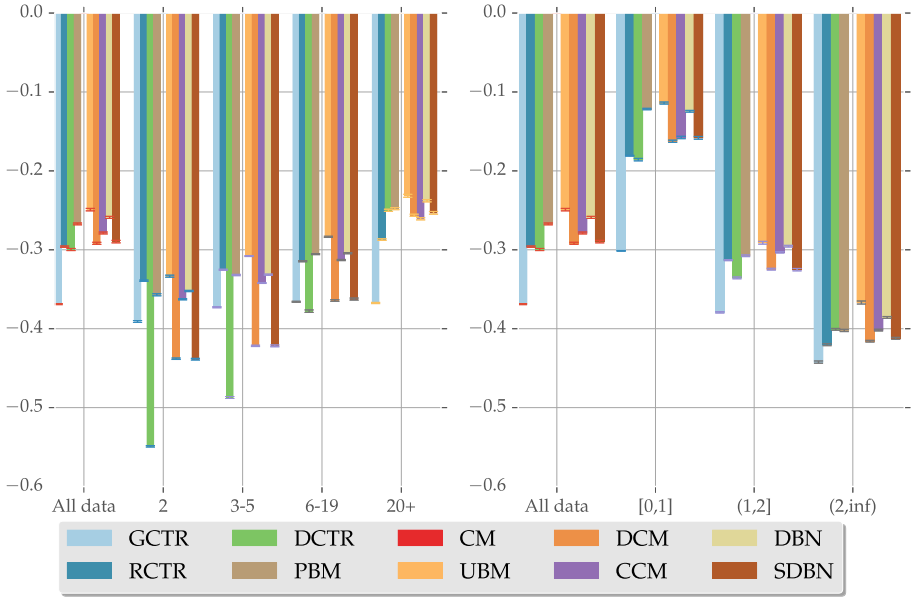

Fig. 1. Log-likelihood of click models, grouped by query frequency (left) and click entropy (right).

contains the evaluation outcomes for every model when trained on the entire dataset.

Log-likelihood. Figure 1 shows the results of the log-likelihood experiments; shorter bars indicate better results. The cascade model (CM) cannot handle multiple clicks in one session and gives zero probability to all clicks below the first one. For such sessions its log-likelihood is $\log 0 = -\infty$ and so the total log-likelihood of CM is $-\infty$.

When evaluated on the whole test set, UBM shows the best log-likelihood, followed by DBN, PBM and CCM. Note that the simplified DBN model (SDBN) has lower log-likelihood values compared to its standard counterpart (DBN). The simple CTR-based models show the lowest log-likelihood. This confirms that complex click models explain and approximate user behavior better than simply counting clicks.

Figure 1 (left) shows the log-likelihood of click models for different query frequencies. In general, the higher the query frequency (more training data available) the better the performance of click models. When comparing complex click models, there is variation in their relative performance based on the query frequency, but UBM consistently has the highest log-likelihood. SDBN and DCM have considerably lower log-likelihood than the similar models DBN and CCM (apart from the "20+" bin). In contrast, the log-likelihood of the CTR-based models varies considerably across query frequencies. On the "2" and "3–5" bins, GCTR outperforms SDBN and DCM, while RCTR is the second best model

overall (after UBM). The DCTR model has the lowest log-likelihood for all query frequencies, but "20+". There, it outperforms SDBN, DCM and CCM and comes close to PBM. These results show two interesting facts. On the one hand, the log-likelihood of complex click models is more stable across different query frequencies than that of the CTR-based models. On the other hand, for each query frequency bin there is a CTR-based model that has log-likelihood scores comparable to complex models (RCTR for "2–19" and DCTR for "20+").

Figure 1 (right) shows the log-likelihood of click models for queries with different click entropy. In general, the lower the click entropy the easier it is to approximate clicks and, hence, the better the performance of click models. The relative log-likelihood of different click models for different values of click entropy is similar to that for different query frequencies: UBM is followed in different orders by DBN, PBM and CCM; SDBN and DCM have lower log-likelihood than the above; the log-likelihood of the CTR-based models varies across bins (RCTR is better than SDBN and DCM on $(1, 2]$, DCTR is comparable to PBM and CCM on $(2, \infty)$). As a future work, we plan to investigate the relation between query frequency and click entropy.

Perplexity. Figure 2 shows the perplexity of the click models; the lower the better. When evaluated on all test sessions, most of the complex click models (apart from CM and CCM) have comparable perplexity, with DBN and SDBN having the lowest one, but not significantly so. The CTR-based models have higher perplexity than the complex models, which again confirms the usefulness of existing click models for web search.

The trends for different query frequencies (Figure 2, left) are similar to those for log-likelihood (Figure 1, left): the variation of perplexity of complex click models is not large (but there are different winners on different bins), while the perplexity of the CTR-based models varies considerably (RCTR has the lowest perplexity overall on "2" and "3–5", DCTR is comparable to other models on "20+"). The trends for different values of click entropy are similar (see Figure 2, right). CM performs poorly in all query classes apart from the $[0, 1]$ entropy bin, which is related to the fact that CM is tuned to explain sessions with one click.

CTR Prediction. Figure 3 shows the impact of query frequency and click entropy on the CTR prediction task. Here, the simple models, RCTR and CM, outperform some of the more complex ones. This is because the intuition of these models is exactly what this task has set out to measure. The average rank of the documents in the training data set is 2.43, i.e., they were usually in some of the top positions. As the RCTR and CM models both perform well on documents that are ranked high, this high average rank influences the observed performance. The top performers on this task are sDBN and DCM. It is not clear why there is such a notable gap in performance between DBN and sDBN on this task; it could be speculated that DBN relies more on the satisfactoriness parameters that are not used in this task. Both UBM and PBM have poor performance on this task, we hypothesize that they rely even more on the position dependent

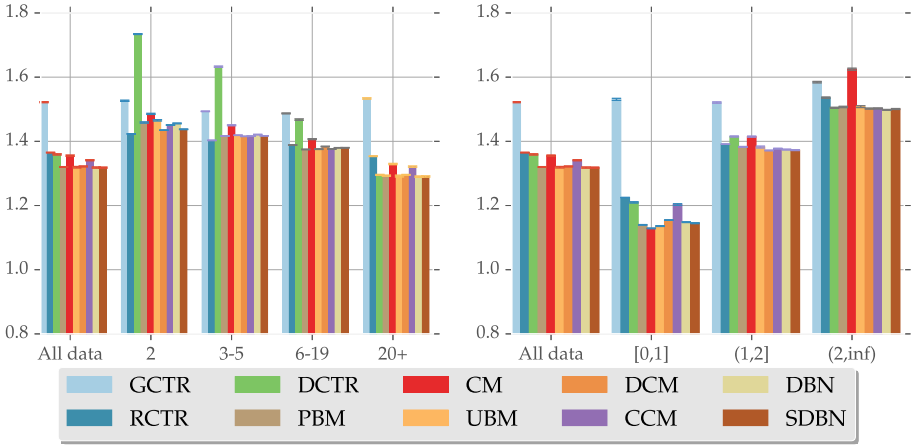

Fig. 2. Perplexity of click models, grouped by query frequency (left) and click entropy (right).

parameters and in this task the document under question was presented at a different position.

Relevance Prediction. The results of the relevance prediction task can be seen in Figure 4. The plot for different query frequencies could not be generated, because the queries with judged results do not occur often in the dataset, while the relevance prediction protocol only considers queries that occur at least ten times.

Table 4. Performance of click models according to various measures: log-likelihood ($\mathcal{LL}$), perplexity, RMSE of the CTR prediction task, AUC of the relevance prediction task, Pearson correlation between annotated relevances and predicted relevances, ranking performance (NDCG@5), and computation time. The symbol ▲ denotes a significant difference at $p = 0.01$ as measured by a two tailed t-test.

Model	$\mathcal{LL}$	Perplexity	RMSE	AUC	Pearson Correlation	NDCG@5	Time (sec.)
GCTR	-0.369	1.522	0.372	0.500	0.000	0.676	0.597
RCTR	-0.296	1.365	0.268	0.500	0.000	0.676	**0.589**▲
DCTR	-0.300	1.359	0.261	0.535	0.054	0.743	3.255
PBM	-0.267	1.320	0.354	**0.581**▲	0.128	0.727	34.299
CM	∞	1.355	0.239	0.515	0.024	0.728	4.872
UBM	**-0.249**▲	1.320	0.343	**0.581**▲	**0.130**▲	0.735	82.778
DCM	-0.292	1.322	**0.212**▲	0.516	0.035	0.733	5.965
CCM	-0.279	1.341	0.283	0.541	0.106	**0.748**	521.103
DBN	-0.259	**1.318**▲	0.286	0.517	0.089	0.719	457.694
SDBN	-0.290	**1.318**▲	**0.212**▲	0.529	0.076	0.721	3.916

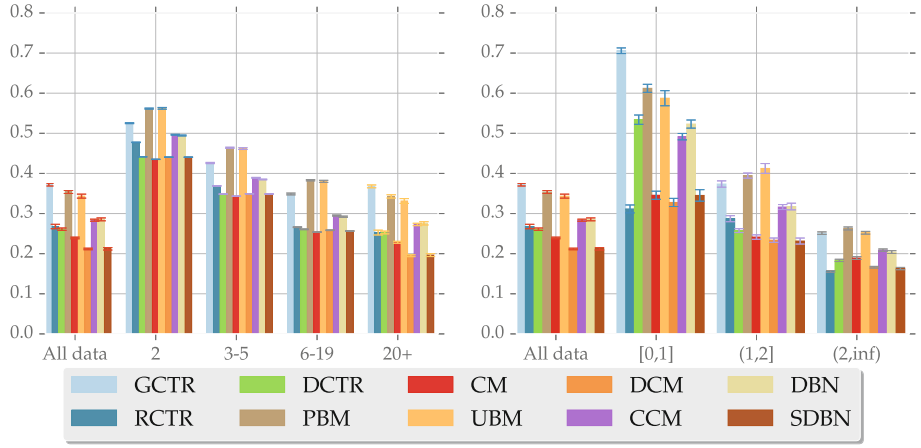

Fig. 3. Click-through rate prediction RMSE of click models, grouped by query frequency (left) and click entropy (right).

The relevance prediction performance of all click models is relatively low (between 0.500 and 0.581). The GCTR and RCTR models do not have a document-specific parameter and, thus, cannot predict relevance. So their AUC is equal to that of random prediction, i.e., 0.5. UBM and PBM have the highest AUC (0.581), while other models are closer to random prediction (from 0.515 for CM to 0.541 for CCM). These results show that existing click models still

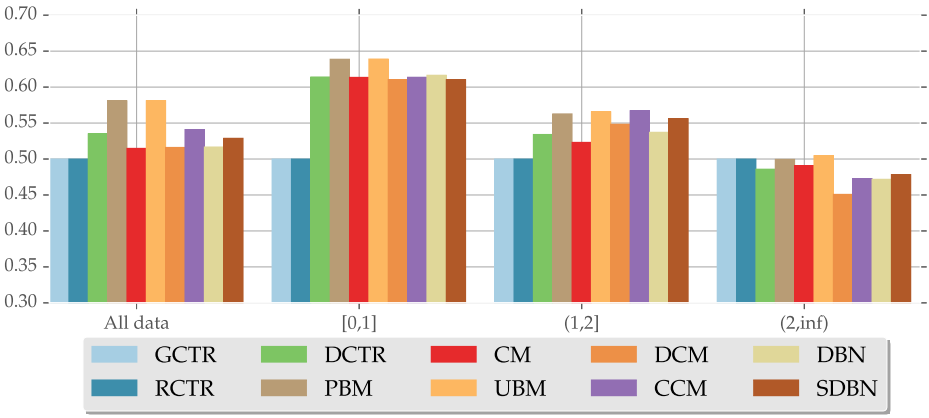

Fig. 4. Relevance prediction of click models on click entropy

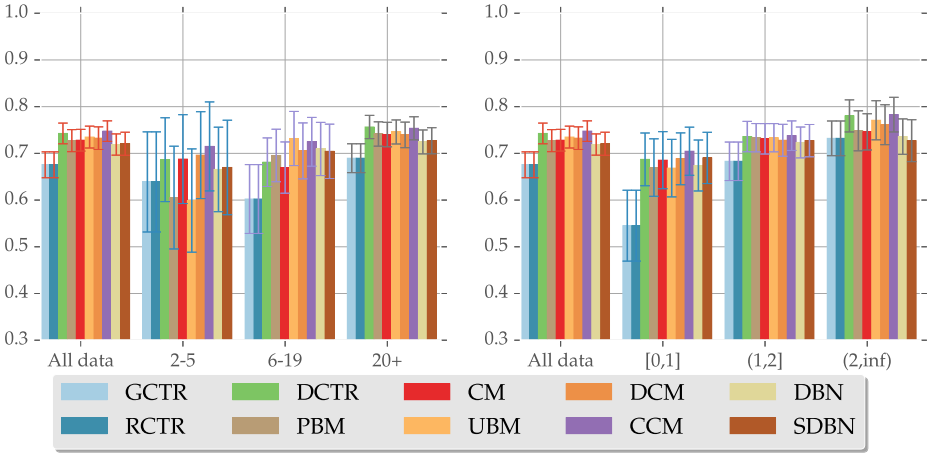

Fig. 5. Ranking performance (NDCG@5) of click models, grouped by query frequency (left) and click entropy (right).

have a long way to go before they can be used for approximating relevance labels produced by human annotators.

Predicted Relevance as a Ranking Feature. Figure 5 shows the results of using the predicted relevance as a ranking feature. The best model here is CCM, followed by the simple DCTR model. This is not surprising as relevant documents attract more clicks and usually have higher CTRs. Thus, ranking documents based on their CTR values only (as done by DCTR) results in high NDCG@5. Notice, though, that predicting actual relevance labels of documents based on the documents' CTRs is still a difficult task (see the discussion above).

The GCTR and RCTR models do not have document-specific parameters and, thus, cannot rank documents. Therefore, they have the lowest values of NDCG@5. They still have high values of NDCG because no reranking was done for documents with equal relevance estimates, hence the values of NDCG for GCTR and RCTR reflect the ranking quality of the original ranker.

Computation Time. In Table 4 we see that, as expected, the models that use MLE inference are much faster than those with EM inference. When using EM inference to calculate the parameters of a click model, one would ideally use some convergence criteria; we have chosen to do a fixed number of iterations (i.e., 50). Notice that UBM is 5–6 times faster than DBN and CCM, even though they all use EM. DBN and CCM use more complex update rules and this results in such a big difference in training time.

Overall Results. We summarize our experimental results in Table 4. There is no perfect click model that outperforms all other models on every evaluation metric. For example, UBM is the best in term of log-likelihood and relevance prediction, while DBN is the best in terms of perplexity and CTR prediction.

Even simple CTR-based models have relatively high performance according to some metrics (e.g., DCTR according to NDCG@5).

6 Conclusion

We have shown that a universal benchmark is necessary for developing and testing click models. The unified evaluation we performed gave important insights into how click models work. In particular, we found that complex click models dominate most of the evaluation metrics, however, in some cases simple click models outperform state-of-the-art models. We also found that none of the tested click models outperforms all others on all measures, e.g., DBN and sDBN are best when judged by perplexity, UBM is best when judged by likelihood, GCTR and RCTR are the fastest and CCM is best for ranking documents.

Our results suggest that different click models can excel at some tasks while having inferior performance at others. Hence, when introducing a new click model or improving an existing one it is important to keep in mind how it is going to be used. If a click model is going to be used for reranking, then the log-likelihood or the perplexity do not matter as much as the ability of the model to rerank documents, and if a click model is going to be used to understand user behavior, then the reranking performance is less important than its ability to explain observations as measured by log-likelihood and perplexity. It is not clear if a single click model can be designed to cater for all needs. Potentially optimizing the design of a click model to a particular use case may improve performance.

We also showed that considering query frequency and click entropy increases the amount of information that can be gained from click model evaluation. In some of the cases our findings were counter intuitive, e.g., higher query frequency did not always make log-likelihood higher. Also, when ranking models by performance, different rankings are observed depending on query frequency or click entropy. This again suggests that no single model can beat all other and that one may benefit from either designing different models for different settings or using an ensemble of models.

The CTR prediction task seems to mimic the behavior of perplexity at the first rank and as such does not give any additional insights into model performance. Relevance prediction also does not give any new insights, albeit for a different reason, the presence of a large set of unseen document-query pairs when evaluating the models.

Our evaluation only covers some of the many click models that have been proposed. The potential for future work is great in the sense that the same evaluation approach can be applied to other click models.

Acknowledgements. This research was supported by grant P2T1P2_152269 of the Swiss National Science Foundation, Amsterdam Data Science, the Dutch national program COMMIT, Elsevier, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 312827 (VOX-Pol), the ESF Research Network Program ELIAS, the HPC Fund, the Royal Dutch Academy of Sciences (KNAW)

under the Elite Network Shifts project, the Microsoft Research PhD program, the Netherlands eScience Center under project number 027.012.105, the Netherlands Institute for Sound and Vision, the Netherlands Organisation for Scientific Research (NWO) under project nrs 727.011.005, 612.001.116, HOR-11-10, 640.006.013, 612.066.930, CI-14-25, SH-322-15, and the Yahoo! Faculty Research and Engagement Program.

All content represents the opinion of the authors which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

1. Chapelle, O., Metzler, D., Zhang, Y., Grinspan, P.: Expected reciprocal rank for graded relevance. In: CIKM 2009, pp. 621–630 (2009)
2. Chapelle, O., Zhang, Y.: A dynamic bayesian network click model for web search ranking. In: WWW 2009, pp. 1–10 (2009)
3. Chuklin, A., Markov, I., de Rijke, M.: Click Models for Web Search. Morgan & Claypool (2015)
4. Chuklin, A., Serdyukov, P., de Rijke, M.: Click model-based information retrieval metrics. In: SIGIR 2013, pp. 493–502 (2013)
5. Craswell, N., Zoeter, O., Taylor, M., Ramsey, B.: An experimental comparison of click position-bias models. In: WSDM 2008, pp. 87–94 (2008)
6. Dou, Z., Song, R., Wen, J.R., Yuan, X.: Evaluating the effectiveness of personalized web search. IEEE TKDE **21**(8), 1178–1190 (2009)
7. Dupret, G., Liao, C.: A model to estimate intrinsic document relevance from the clickthrough logs of a web search engine. In: WSDM 2010, pp. 181–190 (2010)
8. Dupret, G.E., Piwowarski, B.: A user browsing model to predict search engine click data from past observations. In: SIGIR 2008, pp. 331–338 (2008)
9. Guo, F., Liu, C., Kannan, A., Minka, T., Taylor, M., Wang, Y.M., Faloutsos, C.: Click chain model in web search. In: WWW 2009, pp. 11–20 (2009)
10. Guo, F., Liu, C., Wang, Y.M.: Efficient multiple-click models in web search. In: WSDM 2009, pp. 124–131 (2009)
11. Hofmann, K., Schuth, A., Whiteson, S., de Rijke, M.: Reusing historical interaction data for faster online learning to rank for IR. In: WSDM 2013, pp. 183–192 (2013)
12. Hofmann, K., Whiteson, S., de Rijke, M.: A probabilistic method for inferring preferences from clicks. In: CIKM 2011, pp. 249–258 (2011)
13. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems (TOIS) **20**(4), 422–446 (2002)