Click model evaluations
(IR2 project)

Luka Stout and Finde Xumara

University of Amsterdam

1 Introduction

Modeling user behavior on a search engine result page is important for understanding users and supporting simulation experiments. As result pages become more complex, click models have to evolve as well in order to capture additional aspects of user behavior in response to new forms of result presentation. In recent years many models have been proposed that are aimed at predicting behaviour of web search users.

In this project, we implement and evaluate different click models using multiple evaluation metrics and report on the performance of these click models. The click models include the Click-Through Rate click model (CTR), Position-Based click model (PBM), Cascade click model (CM) [9], User Browsing Model (UBM) [4], Click Chain Model (CCM) [5] and Task-centric Click Model (TCM) [12]. The evaluation metrics we use to evaluate performance of these click models are loglikelihood, perplexity, click-through-rate prediction, relevance prediction, ranking performance and computation time. We will also analyze two different factors that might influence performance of the click models, query frequency and click entropy. By doing this experiment we will know the performance of each different click model and this information can be helpful in the creation of new click models and used as a performance benchmark of new click model proposals.

This report is organized as follows. In Section 2 we will describe different click models with their inference. Section 3 will cover the evaluation algorithms used in our experiments. The experiments and data source information will be covered in Section **??**, followed by the results of these experiments.

2 Click Models

In this section, we briefly describe their main characteristics and differences. We have implemented CTR, PBM, CM, CCM, and TCM ourselves, the UBM algorithm is taken from PyClick [11]. The general notation used in this report will be as follows:

Symbol	Description
u	Document u
q	Query q
A	Attractiveness variable
α	Attractiveness parameter
S	Satisfaction variable
σ	Satisfaction parameter
E	Examination variable
ϵ	Examination parameter
R	Relevance variable
r	Relevance parameter
C	Click variable
c	Actual click
$\mathcal{S}$	All sessions
s	A query session
j	Position within a query session

Table 1: Notations used in the equations

2.1 Click-through Rate click model

The most simple click model, Click-Through Rate (CTR), tries to predict relevance for a query-document pair. Relevance is the only parameter in this model which is the click-through rate formula. The inference is done by using maximum likelihood estimation (MLE) because there are no latent variables in this model. Therefore, the click probability of document at position u will be:

$$\begin{aligned} P(C_{uq} = 1) &= P(R_{uq} = 1) \\ P(R_u = 1) &= \frac{1}{|S_{uq}|} \sum_{s \in S_{uq}} c_{uq} \\ \text{where}, S_{uq} &= \{s_q : u \in s_q\} \end{aligned} \tag{1}$$

2.2 Position based model

This model builds upon the CTR model. It adds a `position bias` where documents in a higher position are examined more often. A document can only be clicked if it was examined. The click probability of a document u is as is defined in Equation 2 where $P(R_{uq} = 1)$ is defined as in Equation 1. The values of $P(E_r = 1)$ will be fixed. They are taken from [8] and are $[.68, .61, .48, .34, .28, .2, .11, .1, .08, .06]$. Inference of this model is done by using expectation-maximization (EM)

$$P(C_{uq} = 1) = P(E_{j_u} = 1)P(R_{uq} = 1) \tag{2}$$

where j_u is the rank of document u.

2.3 Cascade model

Cascade model (CM) is another extension to the CTR model. It assumes that users abandon a search session after the first click and hence does not provide a complete picture of how multiple clicks arise in a query session and how to estimate document relevance from such data [9,2]. This model introduces the *cascade hypothesis*, a used examines a search result page (SERP) from top to bottom, deciding whether to click each result before moving to the next. This means ofcourse that to make a click, the user must have decided both to click and skip the ranks above. The relevance of a document is calculated in the same way as in Equation 1. The click probability is defined as:

$$\begin{aligned} P(C_{uq} = 1) &= P(E_{j_u} = 1)P(R_{uq} = 1) \\ P(E_{j_u} = 1) &= \prod_{i=1}^{j_u - 1} 1 - P(C_{i_u} = 1) \end{aligned} \tag{3}$$

The inference of the parameters of CM is done using MLE-inference.

2.4 User Browsing Model

In [4], Dupret and Piwowarski propose a new click model called the User Browsing Model (UBM). The main difference between UBM and other models is that UBM takes the distance into account from the current document u_j to the last clicked document $u_{j'}$ for determining the probability that the user continues browsing:

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

A graphical representation of the model is presented in Figure 1.

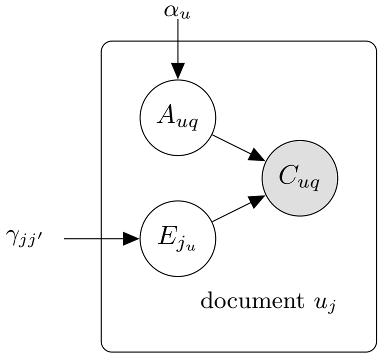


Fig. 1: The graphical model of UBM. PLACE THIS SIDE BY SIDE WITH CCM

2.5 Click Chain Model

In [5], Fan Guo et al, proposed a bayesian based click model which was based on position bias assumption, named Click chain model (CCM). CCM has the following assumptions, some of which are shared with the cascade model.

1. Users are homogeneous: their information needs are similar given the same query;
2. Decoupled examination and click events: click probability is solely determined by the examination probability and the document relevance at a given position;
3. Cascade hypothesis: examination is in strictly sequential order with no breaks.

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 abandonthe search after a while. This model can be formalized with the following conditional probabilities:

$$\begin{gathered}
P(C_{uq}=1|E_{j_u}=0)=0\\
P(C_{uq}=1|E_{j_u}=1,R_{uq})=R_{uq}\\
P(E_{j_u+1}=1|E_{j_u}=0)=0\\
P(E_{j_u+1}=1|E_{j_u}=1,C_{uq}=0)=\tau_1\\
P(E_{j_u+1}=1|E_{j_u}=1,C_{uq}=1,R_{uq})=\tau_2(1-R_{uq})+\tau_3 R_{uq}
\end{gathered} \tag{4}$$

As τ_2 and τ_3 are only used in Equation 4 and in cojunction with eachother and the relevance of the previous document we have decided to combine them into one parameter, τ_{click} and τ_1 has been renamed to τ_{no_click}.

A graphical representation of the model is presented in Figure 2.

2.6 Task-centric Click Model

The Task-centric Click Model (TCM) was first proposed by Zhang et al. in [12]. In the paper they propose a new click model which can handle multiple clicks of multiple queries in a task by introducing two new biases. The first bias indicates that users tend to express their information needs incrementally in a task, thus perform more clicks as their needs become clearer. The other bias indicates that users tend to click fresh documents that are not included in the results of previous queries. In their paper, they named the first assumption as `query bias`, and the second assumption as `duplicate bias`. A graphical representation of the state-of-the-art of the model is presented in Figure **??** and the notations used in TCM are described in Table 2.

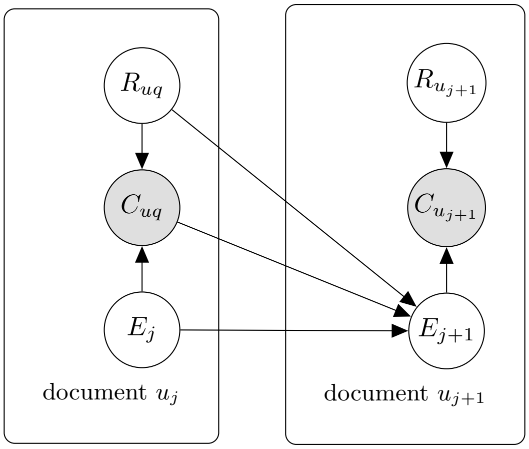


Fig. 2: The graphical model of CCM. PLACE THIS SIDE BY SIDE WITH UBM

Symbol	Description
M_q	Whether the query q matches the user's intent.
N_q	Whether the the user submits another query after q session.
H_{uq}	Previous Examination of the document u in query q.
F_{uq}	Freshness of the document u in query q.
$u'q'$	q' is the latest query session where u has appeared in previous query sessions

Table 2: Additional notations used in TCM

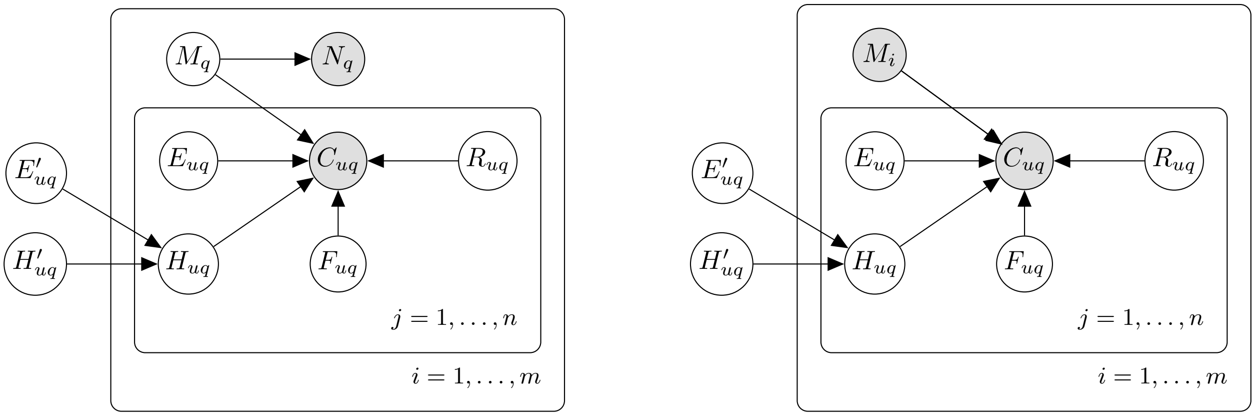
This model can be formalized with the following conditional probabilities:

$$P(M_q = 1) = \tau_1 \tag{5}$$
$$P(N_q | M_q = 1) = \tau_2 \tag{6}$$
$$P(F_{uq} = 1 | M_{uq} = 1) = \tau_3 \tag{7}$$
$$P(E_{uq} = 1) = \epsilon_j \tag{8}$$
$$P(R_{uq} = 1) = r_u \tag{9}$$
$$M_q = 0 \Rightarrow N_i = 1 \tag{10}$$
$$H_{uq} = 0 \Rightarrow F_{uq} = 1 \tag{11}$$
$$H_{uq} = 0 \Leftrightarrow H_{u',q'} = 0, E_{u',q'} = 0 \tag{12}$$
$$C_{uq} = 1 \Leftrightarrow M_q = 1, E_{uq} = 1, R_{uq} = 1, F_{uq} = 1 \tag{13}$$

In our implementation, we simplified TCM model by assuming that a query matches a users intent (M_q) is observed from the click log data, if a user clicks in a result in a query then that query matched the users intent. thus eq.6 can be removed. Our second assumption is that M_q, E_{uq}, R_u and F_{uq} are independent. The detailed calculation for updating EM parameters of the simplified TCM can be found in the appendix. The graphical model of our TCM implementation is presented in Figure 3b.

3 Evaluation Measures

To equally evaluate each click model's performance, we use evaluation metrics that have been proposed in the papers accompanying the proposals of these click models . The evaluation metrics used in this experiment are listed below:



(a) State-of-the-art TCM

(b) Simplified TCM

Fig. 3: The graphical model of state-of-the-art TCM and simplified TCM.

3.1 Loglikelihood

Loglikelihood is the default evaluation metric in machine learning. It says something about the likelihood of the data give the model. In Equation 14 the calculation of the loglikelihood of a click model and a set of sessions can be seen, where $\mathcal{LL}(S|\mathbf{M})$ is the loglikelihood of the sessions given the models parameters, S the set of sessions, $\mathbf{M}$ the model and its parameters, r is the rank of a particular document in a session and $c_r^{(s)}$ a indicator function that is 1 when the document at rank r in session s was clicked and 0 otherwise.

$$\mathcal{LL}(S|\mathbf{M}) = \frac{1}{|S|} \sum_{s \in S} \frac{1}{|s|} \sum_{r=1}^{|s|} \log P(C_r = c_r^{(s)} | \mathbf{M}, C_{r-1}, C_{r-2} \ldots C_1) \tag{14}$$

3.2 Perplexity

Click perplexity is a widely used metric for evaluating click model accuracy. It measures how surprised a model is to see $c_r^{(s)}$ under the current parameters. Perplexity is calculated for every rank individually, as to see whether some models perform better on documents higher on the SERP page then documents ranked lower on the SERP. It is used as a evaluation metric in [12] and [4]. The calculation of perplexity can be seen in Equation 15.

$$Perplexity_r = 2^{-\frac{1}{|S|} \sum_{s \in S} (c_r^{(s)} \log_2 p_r + (1 - c_r^{(s)}) \log_2 (1 - p_r))} \tag{15}$$

$$p_r = P(C_r = 1 | \mathbf{M}) \tag{16}$$

The perplexity of a model is defined as the average of perplexities over all positions. Thus, a smaller perplexity value indicates a better consistency between the click model and the actual click data.

3.3 Click-Trough-Rate Prediction (CTR)

The purpose of click-through rates is to measure the ratio of clicks to impressions of an document. Generally the higher the CTR the higher chance of that document being clicked. The click-through rate of a document d is defined as:

$$CTR_d = \frac{1}{|S_d|} \sum_{s_d} c_{r_d}^{(s_d)} \tag{17}$$

where S_d is the set of sessions where document d appears. A way to use this as an evaluation measure is proposed in [1, p. 4]. In the same way we calculate the CTR prediction using the following protocol:

1. Retrieve all sessions related to a given query.
2. Consider an url that appears both in position 1 and some other positions.
3. Hold out as test sessions all the sessions in which that url appeared in position 1.
4. Train the model on the remaining sessions and predict the relevance.
5. Compute the test CTR in position 1 on the held-out sessions.
6. Compute an error between these two quantities.
7. Average the error on all such urls and queries, weighted by the number of test sessions.

The error measure we use is the Root-Mean-Square-Error (RMSE).

3.4 Relevance Prediction

Relevance prediction was used to evaluate performance of the DBN model [1, p. 6]. The accuracy of CTR prediction may not directly translate to relevance, especially when we were to evaluate the whole task instead of a single query. In this case, the CTR of a particular document is highly dependent on the user-model assumptions. For example if a user tends to ignore a document that isn't fresh, the CTR will be low even if the document is relevant. To measure relevance prediction we use a hand annotated set of relevances. This set contains for a group of query-document pairs a relevance. For these pairs we use the models to predict the relevance. We then use the Area Under the Curve (AUC) between the annotated relevances and the predicted relevances as an evaluation measure. We also calculate the pearson-correlation between the two.

3.5 Predicted Relevance as a Ranking Feature

In this set of experiments we use the predicted relevance directly to rank urls, we use the model as a ranker. To evaluate the performance of a ranker we use the Normalized Discounted Cumulative Gain (NDCG) [7], for which we use a cutoff at five (NDCG@5). To calculate the NDCG@5 we only consider the documents for which we have annotated relevances. All these queries are then averaged to calculate the ranking performance of the click model. The algorithm can be seen below:

1. Retrieve all session that appear more than 10 times.
2. Filter out the sessions that don't appear in the editorial judgements.
3. Train the model on the sessions and predict relevance for the sessions.
4. Sort the urls w.r.t the predicted relevance given by the model.
5. Compute the NDCG@5.
6. Average over all sessions.

3.6 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, we are mostly restricted by the time that it takes to learn a model from the large amount of data that we currently have. This make the ability to efficiently compute parameters an important feature of a succesful model. Therefore we also decided to look at the computation time it takes to train the click models.

4 Experiments

In this section we report on the experimental evaluation and comparison of the click models in Section 2 on the evaluation metrics in Section 3 evaluated on the first 1 million query sessions from the Yandex Relevance Prediction contest of 2011 **??**. The rest of this section will elaborate on the experiments we have done and the results we found. We will also show how query frequency improves the performance of the models. Moreover, we will report on the evaluation measures in combination with click entropy.

4.1 Experimental Setup

The Yandex data set was used in the Yandex Relevance Prediction contest of 2011. In the experiments we have done we have used the first 1 million queries. We deliberatly have chosen to keep sessions that are without any clicks because removing them might hurt the performance of certain models, i.e. TCM because the freshness of a document can change with these sessions. These first 1 million sessions contain 450,931 distinct queries. These queries get 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. The problem with this approach in combination with the TCM model is that this removes the guarantee that the complete task is used when training or evaluating the model. We expect that this will hurt the performance of the TCM model and that the actual performance might be higher.

Performance impacting factors To report on the performance with regard to query frequency we have split the data into 4 parts. The distribution of these parts can be seen in Table 3.We see that there are a lot of queries that are only seen once, these queries are not informative for the models that only look at relevance, e.g. the CTR and CM models, however they are informative for the other models to infer other parameters than relevance. We also see that there are few queries In the rest of this section with every evaluation measure we will also report on how this was influenced by the query frequency. We did this by only using the sessions for a particular group of query frequencies. This way all parameters can be optimal for these frequencies. This does raise a problem however for the query frequency bin of 6 to 19, here there are so few queries it is hard for the models to learn the correct parameters. This will have a negative effect on the performance of the models.

The second factor that might influence performance that we have analyzed is the click entropy. The click entropy was also used to analyse queries in [3].The formal definition is shown in Equation 18.

$$\mathit{ClickEntropy}(q) = \sum_{d\in\mathcal{P}(q)} -P(d|q)\log_2 P(d|q) \tag{18}$$

$$P(d|q) = \frac{\sum_p c^{(q)}_{r_d}}{\sum_{u\in\mathcal{P}(q)} c^{(q)}_{r_u}} \tag{19}$$

Here $\mathit{ClickEntropy}(q)$ is the click entropy for query q. $\mathcal{P}(q)$ are documents clicked on when regarding query q. $P(d|q)$ is the percentage of clicks on document d among all clicks on q. Click entropy can be used to split the data into informational and navigational 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 a informational query the users explore different results to find the optimal one because they don't know what document they are looking for yet. This will give these queries a high click entropy. The search sessions have been divided into 3 parts with respect to click entropy and the evaluation measures have been tested on these parts in the same way as query frequency. Statistics of these parts can be seen in Table **??**.

Query frequency	Number of sessions
1	392508
2 - 5	123799
6 - 19	63596
20 +	420097

Table 3: The distribution of session with respect to query frequency

Click entropy	Number of sessions
0 - 1	509062
1+ - 2	154672
2+	336266

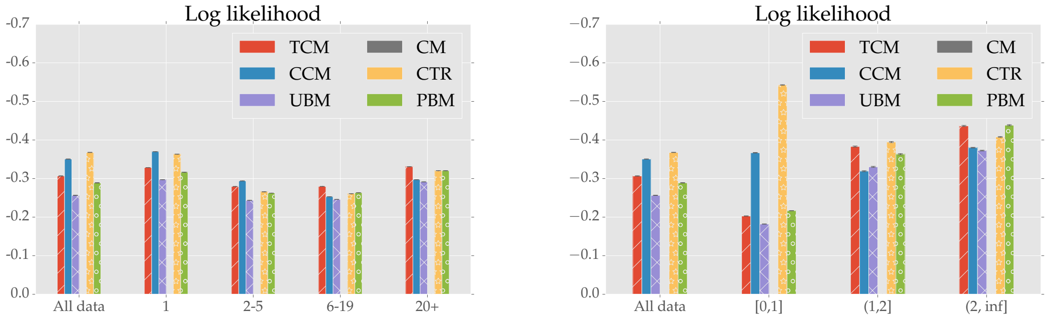
Table 4: The distribution of session with respect to click entropy

4.2 Results

In this section we outline the results of the experiments. For every evaluation measure the influence of the query frequency and click entropy can be seen. Table 5 contains the values of the evaluation measures for every model when trained on the entire dataset.

Loglikelihood In Figures 4a and 4b the results of the loglikelihood experiments can be seen. We want the loglikelihood to be as close to 0 as possible. The loglikelihood of the Cascade Model is not measured as the likelihood of a session with multiple clicks is undefined for that model. We see that UBM is the model that performs best when training on the whole data set and only on the queries with average click-entropy, between 1 and 2, the model is outperformed by CCM. We expected UBM to be the best model going into these experiments. Something to note is that the PBM performs well on queries that have low entropy, a sign that the examination parameters are more accurate on informational queries.

An interesting thing to see is that the loglikelihood gets more negative when looking at the queries with 20+ sessions. We think this is because queries that happen more frequently will also have more documents that are shown to the user overall. This means that a particular document is shown less to the model which makes it harder to predict the click probability.



(a) Query frequency

(b) Click entropy

Fig. 4: Loglikelihood of click models

Perplexity In Figures 6a and 6b the influence of the query frequency and click entropy on the perplexity can be seen. The fact that TCM and PBM perform better than UBM is surprising considering the results of the loglikelihood-experiments. When we compare the figures of loglikelihood (4a, 4b) and perplexity(6a, 6b) with eachother we can see that UBM performs best on loglikelihood while not being the best on perplexity. This is due to the fact that the loglikelihood is the conditional probability of a click, seen in Equation 14 while perplexity uses the unconditional probabilities, seen in Equation 16. UBM uses the distance between the current document and the last click to determine the examination probability which is ofcourse easier to compute if the last click is observed instead of marginalizing over all possible distances since the last click.

In Figure **??** the perplexity of different ranks can be seen. As a document has a higher rank the models perplexity gets lower. However speed with which it gets lower differs per model. TCM and PBM have a high perplexity for the lower ranks, the documents that are more often examined by the users. Whereas for high ranks they are the top performers in terms of perplexity. What this shows is that position based click models, of which our implementation of TCM is one, the examination parameter is more accurate than the examination parameter inferred by other models.

CTR-Prediction Figures 7a and 7b show the impact of query frequency and click entropy on the CTR-prediction task. In this task the simple models, CTR and CM, outperform the more complex ones. This is because the intuition of these models is exactly what this task has set out to measure. As seen in the

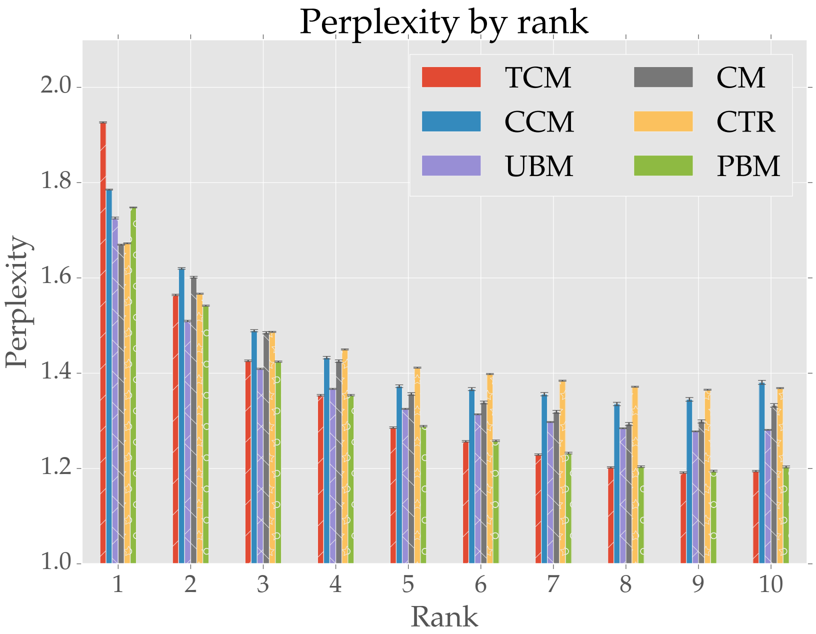
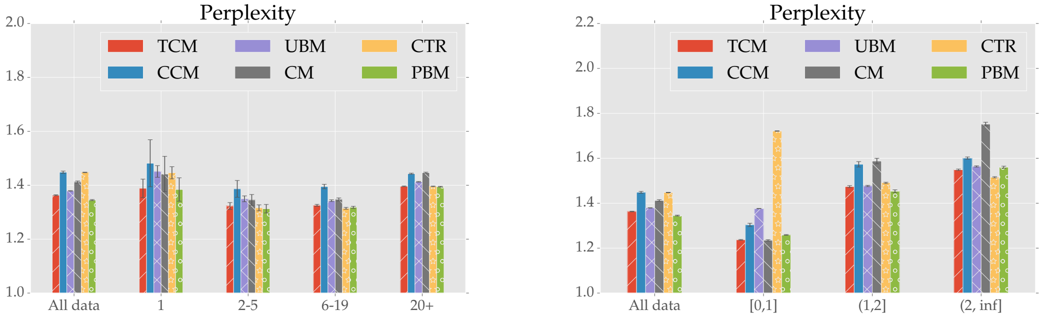


Fig. 5: Perplexity of click models at different document rank position

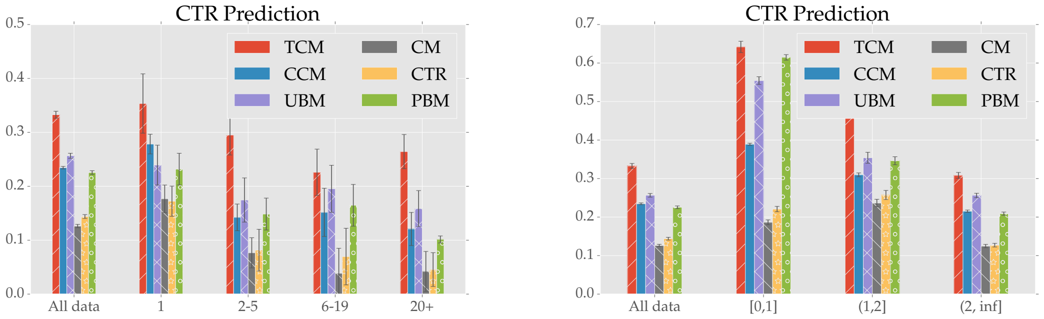


(a) Query frequency

(b) Click entropy

Fig. 6: Perplexity of click models

explanation of this task in Section 3.3 this task is designed to take away the position bias by only testing on the sessions where the document appears in the first place. However it is dissappointing that the complex models perform so much worse. This may have to do with the fact that every time the models are trained on a very small sample of the entire dataset. A consequence of this is that the other parameters, that are used by the complex models besides the relevance parameter, have little data to train on which influences the relevance parameter. Because of this we propose to change the method for calculating the CTR-prediction. By not training on one query at a time but on all queries in the same time the other parameters will train better and thus models that use EM-inference have a better chance to learn the actual relevance of a document.



(a) Query frequency

(b) Click entropy

Fig. 7: Click-through rate of click models

Relevance Prediction The results of the relevance prediction task can be seen in Figure **??**. The plot for query frequency could not be generated for the lower query frequencies because the protocol for calculation only takes queries into account that occur more than 10 times. Even for the click entropy the results for the different splits are not as interesting as all models perform close almost the same. The models are trained only on the small subset of data and are evaluated on all query-document pairs found in the set of annotated relevances a lot of pairs are never seen before by the model. The model will then predict 0.5, whereas the annotated relevance is always 0 or 1. This means that for all unseen query-document pairs the model will get a RMSE of 0.5 and the effect of document-query pairs that are seen will be negligible. When looking at the models that are trained on all the data we can see that the AUC gets lower then 0.5 this means that the predicted relevances have a negative correlation with the annotated relevances, which is illustrated by the Pearson correlation values we found. The values that were found there were between the 0.05 and -0.05, depending on whether the AUC was above or below 0.5. The p-values that we have found were all below 0.05 and indicate that the test is statiscally significant.

Predicted Relevance as a Ranking Feature In Figures 9a and 9b the results of using the predicted relevance as a ranking feature can be seen.

Computation time In Table 5 we see that, as expected, the models that use MLE-inference are much faster than the models with EM-inference. These training times depend on the number of queries that the model is trained upon and the amount of parameters a model has. An interesting thing to note is that the UBM model out performs TCM and CCM by a large margin.

5 Conclusions

In this paper we showed that ...

In our implementation, we did not ...

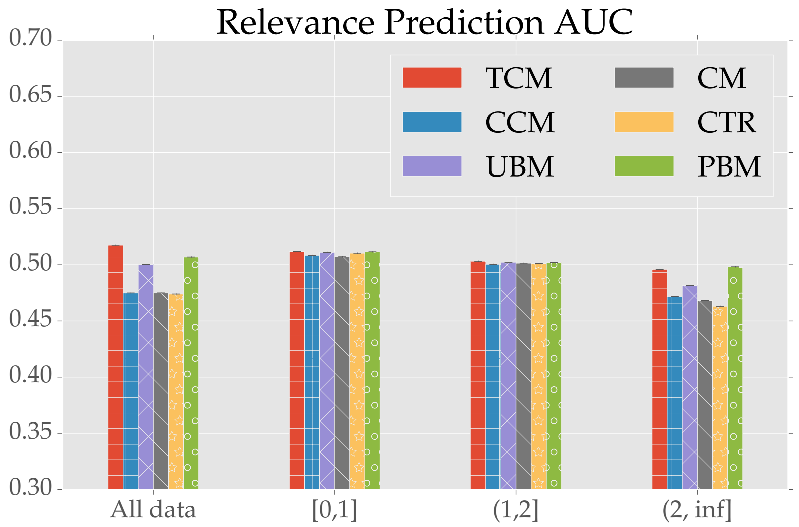
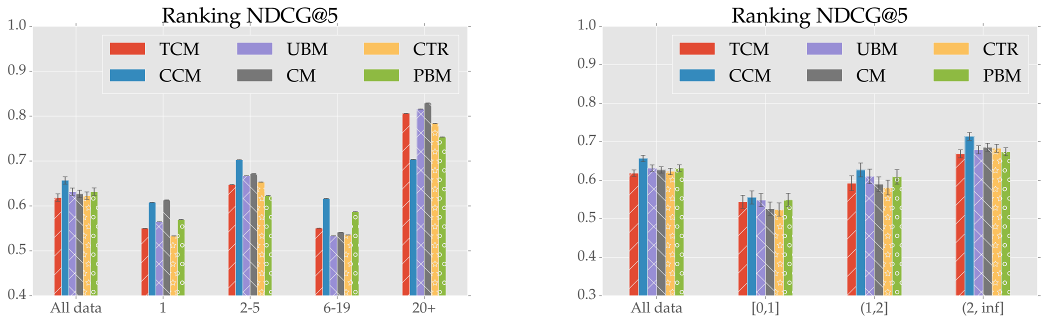


Fig. 8: Relevance prediction of click models on click entropy



(a) Query frequency

(b) Click entropy

Fig. 9: Ranking prediction of click models

Model	Comp.Time.	$\mathcal{LL}$	Perp.	Rel.Pred.AUC	Rel.Pred.COR	Rel.Pred.Pval	Ranking	CTRPred.
TCM	8731.38	-0.307	1.363	0.518	0.0760	2.633e-52	0.619	0.333
CCM	10429.4	-0.3516	1.448	0.475	-0.0475	1.974e-21	0.657	0.234
UBM	5361.18	-0.256	1.379	0.5	0.0159	0.00149	0.632	0.257
CM	139.402	nan	1.412	0.475	-0.0607	6.712e-34	0.627	0.126
CTR	119.683	-0.368	1.448	0.474	-0.0529	3.762e-26	0.623	0.144
PBM	3897.65	-0.289	1.345	0.507	0.0361	5.518e-13	0.632	0.225

Table 5: The results for the experiments on the first million sessions of the dataset. Comp.Time. is Computation time, $\mathcal{LL}$ is the loglikelihood. Perp. is the average perplexity over all ranks. Rel.Pred.AUC is the AUC of the relevance prediction task. Rel.Pred.COR and Rel.Pred.Pval are the Pearson correlation and p-value. Ranking is the NDCG@5 for the ranking performance and CTRPred. is the RMSE of the CTR-prediction task.

References

1. Olivier Chapelle and Ya Zhang. A Dynamic Bayesian Network Click Model for Web Search Ranking Categories and Subject Descriptors. *Www*, pages 1–10, 2009.
2. Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. 2008.
3. Zhicheng Dou, Ruihua Song, Ji-Rong Wen, and Xiaojie Yuan. Evaluating the effectiveness of personalized web search. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2008.
4. Georges E Dupret and Benjamin Piwowarski. A user browsing model to predict search engine click data from past observations. *Sigir '08*, pages 331–338, 2008.
5. Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Min Wang, and Christos Faloutsos. Click chain model in web search. *Proceedings of the 18th international conference on World wide web - WWW '09*, page 11, 2009.
6. Fan Guo, Chao Liu, and Yi Min Ym Wang. Efficient multiple-click models in web search. *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 124–131, 2009.
7. K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.
8. Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 154–161, New York, NY, USA, 2005. ACM.
9. David Kempe and Mohammad Mahdian. A cascade model for externalities in sponsored search. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5385 LNCS:585–596, 2008.
10. Yandex LLC. Task and datasets, n.d.
11. Ilya Markov. Click models for web and aggregated search, 2015.
12. Yuchen Zhang, Weizhu Chen, Dong Wang, and Qiang Yang. User-click modeling for understanding and predicting search-behavior. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11*, page 1388, 2011.

A Appendix

We give here some details about the inference in our TCM implementation outlined in section 2.6.

A.1 Click probability

For $P(F_{i,j} = 1)$ we introduce a variable $f_{i,j}$, which will be derived later.
By assumption that $M_i, E_{i,j}, R_{i,j}$and$F_{i,j}$ are independent, click probability can be formularize as:

$$\begin{aligned} P(C_{i,j} = 1) &= P(M_i = 1) * P(E_{i,j} = 1) * P(R_{i,j} = 1) * P(F_{i,j} = 1) \\ &= \alpha_1 * \beta_j * r_{i,j} * f_{i,j} \end{aligned} \tag{20}$$

A.2 Probability of the query match user intention

Because we remove equation that depends on α_2, we can now set α_1 as MLE.

$$P(M_i = 1) = \alpha_1$$

A.3 Probability of user submit next query

User submit next query if the query does not match user intention (α_1) or user want to search more.

$$\begin{aligned} P(N_i = 1) &= \frac{1}{|S|} \sum_{i \in S} \mathcal{I}(N_i = 1) \\ &= \frac{q_i}{|S|} \\ &= n_i \end{aligned}$$

q_i is the number of submitted-queries where user submit another query after i-th query session.

$$\begin{aligned} P(N_i = 1 | M_i = 1) &= \alpha_2 \\ &= \frac{P(N_i = 1) - P(N_i = 1 | M_i = 0) P(M_i = 0)}{P(M_i = 1)} \\ &= \frac{n_i + \alpha_1 - 1}{\alpha_1} \end{aligned}$$

A.4 Relevance probability

$$\begin{aligned} P(R_{i,j} = 1) &= r_{i,j} \\ &= \frac{\sum_{q_{i,j} \in S_{i,j}} P(R_{i,j} = 1 | C)}{|S_{i,j}|} \end{aligned}$$

Where $S_{i,j}$ are all sessions (queries) containing the document corresponding with the query i at rank j - document $P(R_{i,j} = 1|C)$ will be derive on eq.21

$$\begin{aligned} P(R_{i,j} = 1 | C) &= \mathcal{I}(C_{i,j} = 1) P(R_{i,j} | C_{i,j} = 1) + \mathcal{I}(C_{i,j} = 0) P(R_{i,j} | C_{i,j} = 0) \\ &= c_{i,j} + (1 - c_{i,j}) \frac{P(C_{i,j} = 0 | R_{i,j} = 1) P(R_{i,j} = 1)}{P(C_{i,j} = 0)} \\ &= c_{i,j} + (1 - c_{i,j}) \frac{P(C_{i,j} = 0 | R_{i,j} = 1) r_{i,j}}{1 - P(C_{i,j} = 1)} \end{aligned} \tag{21}$$

Where $c_{i,j} = 1$ if (i,j) was clicked in the current session. $P(C_{i,j} = 0|R_{i,j} = 1)$ is the chance of no click given that it is relevant.

$$
\begin{aligned}
P(C_{i,j} = 0|R_{i,j} = 1) &= P(C_{i,j} = 0|R_{i,j} = 1, M_i = 1)P(M_i = 1) + P(C_{i,j} = 0|R_{i,j} = 1, M_i = 0)P(M_i = 0) \\
&= \alpha_1 P(C_{i,j} = 0|R_{i,j} = 1, M_i = 1, E_{i,j} = 1)P(E_{i,j} = 1) \\
&+ \alpha_1 P(C_{i,j} = 0|R_{i,j} = 1, M_i = 1, E_{i,j} = 0)P(E_{i,j} = 0) \\
&+ (1 - \alpha_1)P(C_{i,j} = 0|R_{i,j} = 1, M_i = 0, E_{i,j} = 1)P(E_{i,j} = 1) \\
&+ (1 - \alpha_1)P(C_{i,j} = 0|R_{i,j} = 1, M_i = 0, E_{i,j} = 0)P(E_{i,j} = 0) \\
\\
&= \alpha_1\beta_j P(C_{i,j} = 0|R_{i,j} = 1, M_i = 1, E_{i,j} = 1, F_{i,j} = 1)P(F_{i,j} = 1) \\
&+ \alpha_1\beta_j P(C_{i,j} = 0|R_{i,j} = 1, M_i = 1, E_{i,j} = 1, F_{i,j} = 0)P(F_{i,j} = 0) \\
&+ \alpha_1(1 - \beta_j)P(C_{i,j} = 0|R_{i,j} = 1, M_i = 1, E_{i,j} = 0, F_{i,j} = 1)P(F_{i,j} = 1) \\
&+ \alpha_1(1 - \beta_j)P(C_{i,j} = 0|R_{i,j} = 1, M_i = 1, E_{i,j} = 0, F_{i,j} = 0)P(F_{i,j} = 0) \\
&+ (1 - \alpha_1)\beta_j P(C_{i,j} = 0|R_{i,j} = 1, M_i = 0, E_{i,j} = 1, F_{i,j} = 1)P(F_{i,j} = 1) \\
&+ (1 - \alpha_1)\beta_j P(C_{i,j} = 0|R_{i,j} = 1, M_i = 0, E_{i,j} = 1, F_{i,j} = 0)P(F_{i,j} = 0) \\
&+ (1 - \alpha_1)(1 - \beta_j)P(C_{i,j} = 0|R_{i,j} = 1, M_i = 0, E_{i,j} = 0, F_{i,j} = 1)P(F_{i,j} = 1) \\
&+ (1 - \alpha_1)(1 - \beta_j)P(C_{i,j} = 0|R_{i,j} = 1, M_i = 0, E_{i,j} = 0, F_{i,j} = 0)P(F_{i,j} = 0)
\end{aligned}
$$

We note that $P(C_{i,j} = 0|R_{i,j} = 1, M_i = 1, E_{i,j} = 1, F_{i,j} = 1) = 0$. Otherwise it is 1. From eq. 24 from TCM paper. Together with inserting our parameters this gives us the following:

$$
\begin{aligned}
P(C_{i,j} = 0|R_{i,j} = 1) &= \alpha_1\beta_j f_{i,j} + \alpha_1\beta_j(1 - f_{i,j}) + \alpha_1(1 - \beta_j)f_{i,j} + \alpha_1(1 - \beta_j)(1 - f_{i,j}) \\
&+ (1 - \alpha_1)\beta_j f_{i,j} + (1 - \alpha_1)\beta_j(1 - f_{i,j}) + (1 - \alpha_1)(1 - \beta_j)(f_{i,j} \\
&+ (1 - \alpha_1)(1 - \beta_j)(1 - f_{i,j})
\end{aligned}
$$

expanding this we are only left with

$$
P(C_{i,j} = 0|R_{i,j} = 1) = 1 - (\alpha_1\beta_j f_{i,j}) \tag{22}
$$

Which seems intuitive as we assumed that all $M_i, R_{i,j}, E_{i,j}$ and $F_{i,j}$ are independent to get $P(C_{i,j} = 1)$. With this information we can calculate

$$
\begin{aligned}
P(R_{i,j} = 1|C) &= c_{i,j} + (1 - c_{i,j})\frac{(1 - (\alpha_1\beta_j f_{i,j}))r_{i,j}}{1 - \alpha_1\beta_j f_{i,j} r_{i,j}} \\
&= c_{i,j} + (1 - c_{i,j})\frac{r_{i,j} - \alpha_1\beta_j f_{i,j} r_{i,j}}{1 - \alpha_1\beta_j f_{i,j} r_{i,j}}
\end{aligned}
$$

A.5 Examination probability

$$
\begin{aligned}
P(E_{i,j} = 1) &= \beta_j \\
&= \frac{1}{|S|}\sum_{i \in S} P(E_{i,j} = 1|C)
\end{aligned}
$$

Where S is all sessions and i is a query within that session. $P(E_{i,j} = 1|C)$ will be derive on eq.23

$$
\begin{aligned}
P(E_{i,j} = 1|C) &= \mathcal{I}(C_{i,j} = 1)P(E_{i,j}|C_{i,j} = 1) + \mathcal{I}(C_{i,j} = 0)P(E_{i,j}|C_{i,j} = 0) \qquad (23) \\
&= c_{i,j} + (1 - c_{i,j})\frac{P(C_{i,j} = 0|E_{i,j} = 1)P(E_{i,j} = 1)}{P(C_{i,j} = 0)} \\
&= c_{i,j} + (1 - c_{i,j})\frac{P(C_{i,j} = 0|E_{i,j} = 1)\beta_j}{1 - P(C_{i,j} = 1)}
\end{aligned}
$$

Where $c_{i,j}$ indicates whether document i, j was clicked. Analog to eq 22 we can show that

$$P(C_{i,j} = 0|E_{i,j} = 1) = 1 - (\alpha_1 f_{i,j} r_{i,j})$$

This gives us

$$\begin{aligned} P(E_{i,j} = 1|C) &= c_{i,j} + (1 - c_{i,j}) \frac{(1 - (\alpha_1 f_{i,j} r_{i,j}))\beta_j}{1 - \alpha_1 \beta_j f_{i,j} r_{i,j}} \\ &= c_{i,j} + (1 - c_{i,j}) \frac{\beta_j - \alpha_1 \beta_j f_{i,j} r_{i,j}}{1 - \alpha_1 \beta_j f_{i,j} r_{i,j}} \end{aligned}$$

A.6 Freshness probability

$$\begin{aligned} P(F_{i,j} = 1|H_{i,j} = 1) &= \alpha_3 \\ \alpha_3 &= \frac{1}{|S_{i,j}|} \sum_{q \in S} \sum_{(i,j) \in q} P(Fi, j = 1|H_{i,j} = 1, C) \end{aligned} \tag{24}$$

Where (i,j) is a query, rank pair identifying a certain document. $P(F_{i,j} = 1|C)$ will be derived in eq.25 $P(F_{i,j} = 1)$ will be derived in eq.27

$$\begin{aligned} P(F_{i,j} = 1|H_{i,j} = 1, C) &= \mathcal{I}(C_{i,j} = 1) P(F_{i,j} = 1|H_{i,j} = 1, C_{i,j} = 1) \\ &+ \mathcal{I}(C_{i,j} = 0) P(F_{i,j} = 1|H_{i,j} = 1, C_{i,j} = 0) \\ &= c_{i,j} + (1 - c_{i,j}) \frac{P(C_{i,j} = 0|F_{i,j} = 1, H_{i,j} = 1) P(F_{i,j} = 1|H_{i,j} = 1)}{P(C_{i,j} = 0|H_{i,j} = 1)} \end{aligned} \tag{25}$$

Analog to eq 22 we can show that

$$P(C_{i,j} = 0|F_{i,j} = 1, H_{i,j} = 1) = 1 - (\alpha_1 \beta_j r_{i,j}) \tag{26}$$

We can also show

$$\begin{aligned} P(C_{i,j} = 0|H_{i,j} = 1) &= 1 - P(C_{i,j} = 1|H_{i,j} = 1) \\ &= 1 - (\alpha_1 \alpha_3 \beta_j r_{i,j}) \end{aligned}$$

The only difference between this and eq. 20 is that it is given that $H_{i,j} = 1$ and because $H_{i,j} = 1$ only has an influence on $P(F_{i,j} = 1)$, namely that $P(F_{i,j} = 1|H_{i,j} = 1) = 1$, we can substitute $f_{i,j}$ with $\alpha 3$ in eq. 20

Now we only need to calculate $f_{i,j} = P(F_{i,j}) = 1$

$$\begin{aligned} P(F_{i,j} = 1) &= \mathcal{I}(H_{i,j} = 1) P(F_{i,j} = 1|H_{i,j} = 1) + \mathcal{I}(H_{i,j} = 0) P(F_{i,j} = 1|H_{i,j} = 0) \\ &= \mathcal{I}(H_{i,j} = 1)\alpha_3 + \mathcal{I}(H_{i,j} = 0) \end{aligned}$$

Where $\mathcal{I}(H_{i,j} = 1)$ is a binary indicator function from the data specifying whether document (i, j) was shown before in the current (q from eq. 24) session.

We could replace this indicator function with the probability that the document was examined the last time it was shown. This probability, called $H_{i,j}$ would depend on the probability that it was examined and $H_{i',j'}$ where i', j' is the last time this document was shown in the current session. It would look like this

$$P(H_{i,j} = 1) = P(E_{i',j'} = 1) P(H_{i',j'} = 1)$$

then eq. 27 becomes:

$$\begin{aligned}
P(F_{i,j} = 1) &= P(H_{i,j} = 1)\alpha_3 + P(H_{i,j} = 0) \\
&= P(H_{i,j} = 1)\alpha_3 + (1 - P(H_{i,j} = 1)) \\
&= \alpha_3 P(E_{i',j'} = 1)P(H_{i',j'} = 1) + (1 - P(E_{i',j'} = 1)P(H_{i',j'} = 1))
\end{aligned}$$

Note that this discards the information that if (i', j') was clicked it surely was examined.

With eq 26 we can calculate $P(F_{i,j} = 1|C)$

$$\begin{aligned}
P(F_{i,j} = 1|H = 1, C) &= c_{i,j} + (1 - c_{i,j})\frac{(1 - (\alpha_1\beta_j r_{i,j}))\alpha_3}{1 - \alpha_1\alpha_3\beta_j r_{i,j}} \qquad (27) \\
&= c_{i,j} + (1 - c_{i,j})\frac{\alpha_3 - \alpha_1\alpha_3\beta_j r_{i,j}}{1 - \alpha_1\alpha_3\beta_j r_{i,j}}
\end{aligned}$$