# A New Click Model for Relevance Prediction in Web Search

Alexander Fishkov[1] and Sergey Nikolenko[2,3]

[1] St. Petersburg State Polytechnical University jetsnguns@gmail.com
[2] Steklov Mathematical Institute, St. Petersburg, Russia sergey@logic.pdmi.ras.ru
[3] St. Petersburg Academic University, St. Petersburg, Russia

**Abstract.** We present a new click model for processing click logs and predicting relevance and appeal for query–document pairs in search results. Our model is a simplified version of the task-centric click model but outperforms it in an experimental comparison.

**Keywords:** web search, click models, relevance prediction

## 1 Introduction

Search engines process huge amounts of information: the text of billions of web pages and hyperlinks between them that form the structure of the World Wide Web. Obviously, this information, usually provided by web crawlers, lies in the foundation of a successful search engine [1]. However, as a search engine accumulates active users, information about their behaviour begins to weigh in: *click logs* accumulate first-hand information on user behaviour, i.e., which search results for a certain query users *actually click*. Obviously, the best possible relevance estimates come from the humans themselves; thus, click log information represents an invaluable resource on which search engines would like to draw.

In this work, we propose a new model for processing click logs which is simpler for inference than an existing task-centric click model (TCM) but produces better results. In Section 2, we review existing click models and introduce basic definitions and problem setting; a separate Section 3 is devoted a detailed description of TCM. In Section 4, we present our modified click model and describe the inference procedure. Section 5 describes our experimental setup and results produced on a publicly available large-scale dataset, and Section 6 concludes the paper.

## 2 Related work

Recent years have seen a proliferation of click models for modeling user behaviour. This line of research began in studying the *position bias* effect: user behaviour studies have shown [2] that not only higher positions in search results rankings attract more attention and are more likely to be clicked on, but also that lower positions are often not even examined at all by the user. Ensuing

probabilistic modeling confirmed these results and formalized them in the *examination hypothesis* [3] that captures this reasoning by specifying probabilities of the event $C_i$ that the user clicks on document at position $i$ as conditional probabilities on the event $E_i$ that the user actually examines the document at position $i$; the examination hypothesis states that $p(C_i = 1 \mid E_i = 0) = 0$.

Latest research has built upon this assumption and has incorporated various additional assumptions and new pieces of information that could be used to predict the click event. Early models tried to capture position bias directly: the *clicks over expected clicks* model [4] estimates the number of expected clicks for each position, the *examination model* learns position bias with an EM algorithm [5], and logistic regression has also been used to estimate position biases [3].

However, it is actually true that in a good search engine, top results are generally more relevant than bottom results, so position bias is not just a feature of the user's perception as position models presuppose but also has sound underlying causes. Thus, emphasis shifted to more complex probabilistic graphical models that attempt to more accurately model actual user behaviour. They are usually based on the *cascade hypothesis* [3]: a user examines documents from top to bottom, so a document at position $i + 1$ can be examined only if the document at position $i$ has been examined: $p(E_{i+1} = 1 \mid E_i = 0) = 0$. A notable model that does not use the cascade hypothesis is the *user browsing model* (UBM) proposed in [5]. UBM assumes that the user "jumps" from the previously clicked position $i_c$ to one of the subsequent positions $i$ with constant probabilities: $p(E_i = 1 \mid C_{i_c}) = \beta_{i_c, i - i_c}$.

Several graphical click models with varying complexity have been proposed under the cascade hypothesis [6–13]. Starting from the *dynamic Bayesian network* (DBN) model [6], click models usually draw a distinction between *appeal* and *relevance*, or, in terms of [6], *appeal*, *perceived relevance* and *intrinsic relevance*. Appeal shows how relevant the document looks for the user; it is directly responsible for user clicks. Perceived relevance shows how relevant the user has felt the document to be after the user has clicked on it and looked at it; perceived relevance is responsible for user satisfaction and, therefore, for the fact whether the user comes back and examines subsequent documents after this one. Intrinsic relevance is usually an auxiliary feature derived from appeal and perceived relevance: appeal and perceived relevance are usually normalized to lie between 0 and 1, and intrinsic relevance is computed as their product.

One of the latest click models is the *task-centric click model* (TCM) proposed by Zhang et al. [7]. TCM steps back and considers whole *sessions* of queries submitted by the same user, assuming that the user has a certain purpose in mind (hence *task-centric*), and various queries are intended to carry out that purpose. Our model, presented in the next section, is close to TCM in essence but turns out to be better in experimental studies and simpler for inference.

## 3 The TCM model

The main characteristic feature of the task-centric click model is a broader look at search process: interaction between user and search engine is viewed as a sequential process of submitting and reformulating queries. TCM assumes that the user has a specific informational need, a search intent which is assumed to be fixed during the entire session. The user enters a query, examines the result, and then decides whether to click on some documents or enter another query and so on. There are two main assumptions about user behavior in TCM:

(1) if a query does not match the user's underlying intent, he will perform no clicks but learn from search results to pose a new, refined query;
(2) when a document has been examined before in the same session, it will have a lower probability to be clicked when the user examines it again.

For the $i^{\text{th}}$ query in a session and for the $j^{\text{th}}$ document in the search results, TCM introduces the following variables:

$M_i$, whether the $i^{\text{th}}$ query matches the user's intent;
$N_i$, whether the user submits another query after the $i^{\text{th}}$(*observed*);
$E_{i,j}$, whether the user examines the document at $(i, j)$;
$H_{i,j}$, whether the document at $(i, j)$ has already been shown during the current session ;
$F_{i,j}$, whether the document is considered fresh by the user;
$C_{i,j}$, whether the document is clicked (*observed*);
$R_{i,j}$, whether the document is relevant;
$(i', j')$, previous position of document at $(i, j)$ if this document has already been shown during the current session .

The following formulas complete the definition of TCM:

$$p(M_i = 1) = \alpha_1, \qquad p(R_{i,j} = 1) = r_{i,j},$$
$$p(N_i = 1|M_i = 1) = \alpha_2, \qquad p(E_{i,j} = 1) = \beta_j,$$
$$p(N_i = 1|M_i = 0) = 1, \qquad H_{i,j} = 0 \Leftrightarrow H_{i',j'} = E_{i',j'} = 0,$$
$$p(F_{i,j} = 1|H_{i,j} = 1) = \alpha_3, \qquad C_{i,j} = 1 \Leftrightarrow M_i = E_{i,j} = A_{i,j} = F_{i,j} = 1,$$
$$p(F_{i,j} = 1|H_{i,j} = 0) = 1.$$

The TCM model is presented as a Bayesian network on Fig. 1. Variables $F$ and $H$ represent the second assumption on user behavior: the probability to click on the current dicument is affected by probabilities of its previous examinations. Looking back at the formulas, one can find out that additional edges are added to the network based on what documents are shown to the user on each

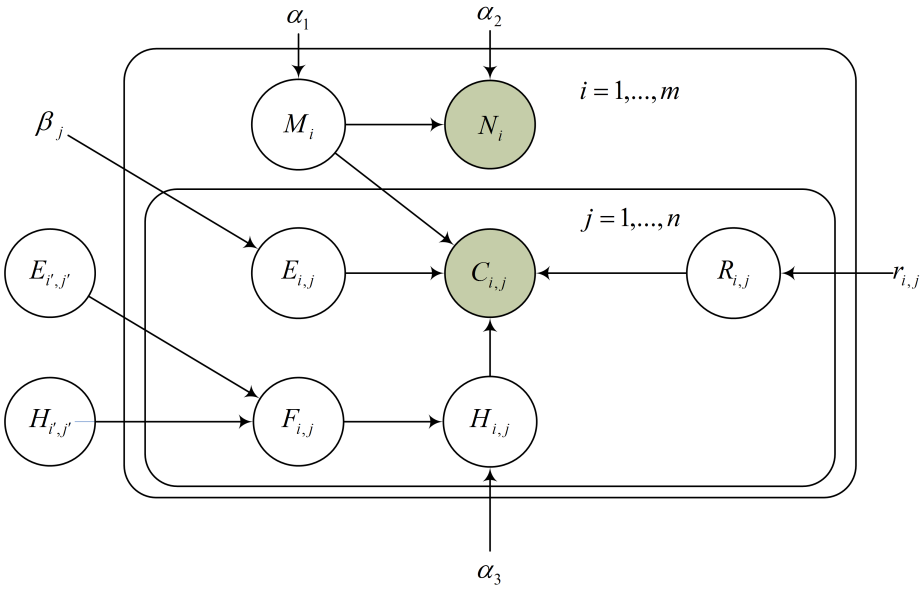


**Fig. 1.** The TCM model. Shaded nodes represent observed variables.

query result page. Our experiments have shown that large sessions with similar queries (some documents are shown many times during session) significantly slow down inference. It would be much better for the model's efficiency to have each query self-contained. Another way to improve the model is to introduce a more complete query-level structure. We tried to adress these points in our model presented in the next section.

## 4 The SCM model

In this work, we propose a new model that we call the *session click model* (SCM). It is essentially a simplification of TCM: we break down some of the connections in the factor graph of TCM in order to ease and speed up Bayesian inference. However, as we will see below, our model actually outperforms TCM on real-world data.

For the $i^{\text{th}}$ query in a session and for the $j^{\text{th}}$ document in the search results, SCM introduces the following variables:

$M_i$, whether the $i^{\text{th}}$ query matches the user's intent;
$N_i$, whether the user submits another query after the $i^{\text{th}}$(*observed*);
$E_{i,j}$, whether the user examines the document at $(i, j)$;
$H_{i,j}$, whether the document at $(i, j)$ has already been shown during the current session (*observed*);
$F_{i,j}$, whether the document is considered fresh by the user;
$C_{i,j}$, whether the document is clicked (*observed*);
$A_{i,j}$, whether the document appeals to the user;
$S_{i,j}$, whether the document satisfies the user.

Unlike TCM, our model uses $H_{i,j}$ as indication of prior appearance of a document so it becomes a new observed variable. In TCM, these variables make up additional connections between different queries in a session; in SCM, they are observed so different queries become dependent only via $M_i$.

Formally, SCM is defined as follows (we write some conditional probabilities as logical formulas for brevity and clarity):

$$
\begin{aligned}
p(M_i = 1) &= \alpha_1, & E_{i,j-1} = 0 &\Rightarrow E_{i,j} = 0,\\
p(N_i = 1 | M_i = 1) &= \alpha_2, & S_{i,j-1} = 1 &\Rightarrow E_{i,j} = 0,\\
p(N_i = 1 | M_i = 0) &= 1, & E_{i,j} = 1 &\Rightarrow E_{i,j-1} = 1 \,\&\, S_{i,j-1} = 0,\\
p(F_{i,j} = 1 | H_{i,j} = 1) &= \alpha_3, & p(S_{i,j} = 1 \mid C_{i,j} = 1) &= s_{i,j},\\
p(F_{i,j} = 1 | H_{i,j} = 0) &= 1, & p(S_{i,j} = 1 \mid C_{i,j} = 0) &= 0,\\
p(A_{i,j} = 1) &= a_{i,j}, & C_{i,j} = 1 &\Leftrightarrow M_i = E_{i,j} = A_{i,j} = F_{i,j} = 1.
\end{aligned}
$$

As a Bayesian network, the model is presented on Fig. 2; the conditional probability tables are shown above, so Fig. 2 together with the above formulas represent a complete specification of the joint probability distribution in SCM. We perform Bayesian inference in SCM via loopy belief propagation. Our model has three global parameters $\alpha_1$, $\alpha_2$, $\alpha_3$. They represent various conditional probabilities. To estimate their values from data, we first convert our model in the form of a factor graph. For each parameter, we add another variable node and connect it to the corresponding factor. Then we iterate through the click log as follows:

(1) assign a uniform Beta prior for each $\alpha_i$;
(2) process a single session from click log and get posteriors for $\alpha_1$, $\alpha_2$, and $\alpha_3$;
(3) for the next session, set priors for them to posteriors from previous session;
(4) return to step 2.

This would be a very lengthy process for the entire click log, but posterior estimates do converge relatively quickly, so we can stop when the variance becomes small enough. Then we use these estimates as parameter values for the SCM.

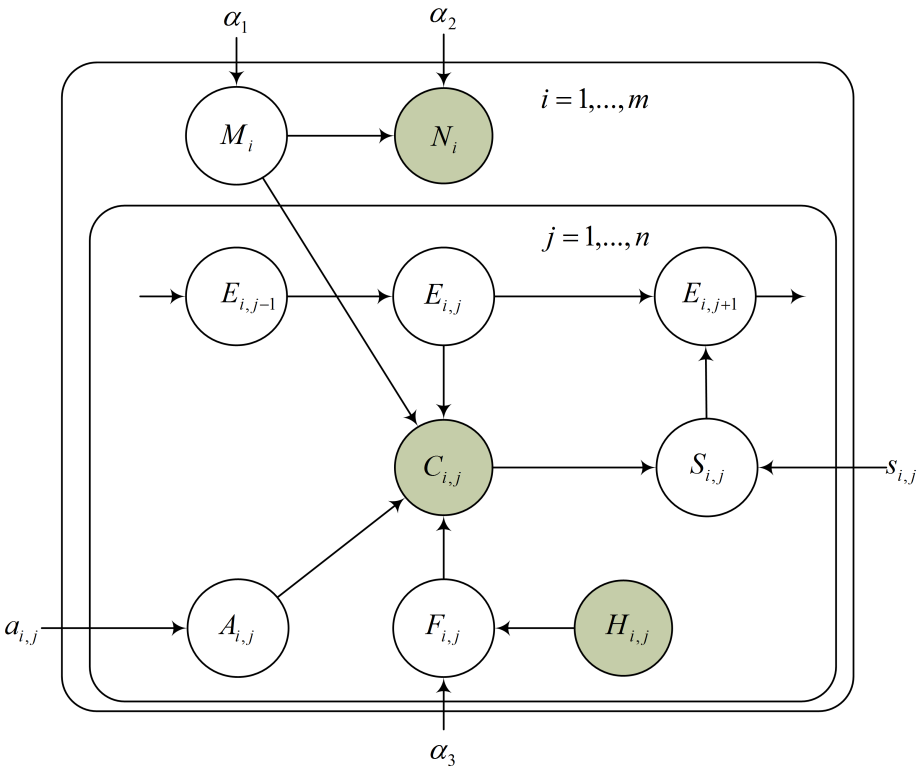


**Fig. 2.** The SCM model. Shaded nodes represent observed variables.

## 5 Experiments

### 5.1 Dataset

We evaluate our results on (a representative subset of) the Yandex click log dataset that was made available for the "Internet Mathematics" competition of 2011 [14]. The data is divided into user sessions that consist of queries, search results, and user clicks for these results; the logs are anonymized, and no user information is provided (we do not know which sessions come from the same user). Click logs also contain time delays between clicks, but neither our model nor any of the competitive click models we compare it with makes use of temporal information; this is a very interesting subject for further study.

### 5.2 Experimental setup

In general, to evaluate the results we use the *area under curve* (AUC) metric [15] computed on a test set with relevances evaluated by experts. AUC is a popular quality metric for classifiers; it represents the probability that for a

uniformly selected pair consisting of a relevant and an irrelevant document the classifier ranks the relevant one higher. Thus, the optimal AUC is 1 (all relevant documents come before irrelevant ones), and a completely random classifier will get, on average, an AUC of 0.5.

However, this is not the whole story. While they do aim to capture user behaviour, click models cannot produce cutting edge results by themselves. To get reasonable prediction accuracy (i.e., a competitive AUC score), click log analysis must also take into account other features that can be inferred from the click logs. For example, one strikingly useful feature is the actual ranking of a document in the original search results: the search engine has processed a lot of additional information which is not available from click logs, and this information has been succinctly represented in the search results rankings, so why not use it. There are many other important features, too.

To simulate this real-world application of click models, we set up our experiments as follows. We have computed 60 *static features* for every query-document pair; these features comprise the base set, and in our experiments, they are augmented by various *dynamic features* that come from click models. We have implemented click models as probabilistic graphical models in the Infer.NET framework [16]; the Infer.NET suite provides readily available inference algorithms, including loopy belief propagation.

To combine all features, static and dynamic, we have used boosting techniques. Following [17, 18], we construct a ranking function by transforming the data into pairwise preferences and considering the resulting problem as a regression problem. To do so, we break every list of search results in the training set into pairwise comparisons between documents. Every ordered pair of documents specified by their features, $f_1$ and $f_2$, is represented by the concatenation of their feature vectors in the same order, $\langle f_1, f_2 \rangle$. Then, the target value is chosen to be 1 if the first document is relevant and the second one is not, $-1$ in the opposite case, and 0 if they are incomparable (both relevant or both irrelevant). These feature vectors together with their training values are fed into a regression boosting algorithm.

In static features, we aimed for simplicity; besides, we did not use the best known boosting techniques [19–21], again choosing a readily available Matlab implementation of least squares regression boosting [22] for simplicity (the learning procedure is stochastic, so we have run it five times and averaged the results). Thus, even the best of our results do not match the top AUC scores obtained in the "Internet Mathematics" competition. For a report of the winners see [23]; the winners did not invent new click models but did everything right with rank boosting and feature generation for other features; see also [24] for a report of a team who used random forests rather than boosting techniques. Nevertheless, we believe that our results do provide a fair comparison in a situation representative of real-life applications.

| Model | Features | AUC |
|---|---|---|
| DBN | appeal | 0.6252 |
| | perceived relevance | **0.6254** |
| | intrinsic relevance | 0.6253 |
| | without static features | 0.5944 |
| TCM | appeal | 0.6255 |
| | perceived relevance | 0.6278 |
| | intrinsic relevance | **0.6279** |
| | without static features | 0.5963 |
| SCM | appeal | 0.6265 |
| | perceived relevance | 0.6294 |
| | intrinsic relevance | **0.6296** |
| | without static features | 0.5964 |
| All models and features together | | **0.6313** |

**Table 1.** Experimental results.

### 5.3 Results

The results are summarized in Table 1. Each click model in our comparison provides three features: appeal, perceived relevance, and intrinsic relevance that we compute as the product of appeal and perceived relevance (an idea first presented in [6]). In SCM, appeal is estimated as the maximum a posteriori estimate of $a_{i,j}$, perceived relevance is estimated as the maximum a posteriori estimate of $s_{i,j}$, and intrinsic relevance is computed as $a_{i,j}s_{i,j}$. We provide results for the ranking resulting from the three features from a single dynamic model alone, without static features, and results of least squares boosting learning on static features together with each of the three dynamic features from a certain model. As we can see, SCM outperforms both DBN and TCM in terms of AUC, both with static features and without them (although different variables come out ahead in different models).

In the last row of Table 1, we also provide the results for all static and all dynamic features from all three models thrown together. Improved AUC suggests that DBN and TCM do capture some aspects of user behaviour that SCM does not; in further work, we plan to investigate this further and bring other sides of user behaviour into our model, hopefully still leaving the model relatively simple.

## 6 Conclusion

In this work, we have proposed a new click log model which is in essence a simplification of the task-centric model but has outperformed it in our experiments. Further work may include extending the model to capture more different aspects of user behaviour (e.g., distinguishing between navigational and informational queries) and devising a large-scale highly parallel implementation of our click model.

**Acknowledgements**

We thank Yandex employees Igor Kuralenok and Andrey Strelkovskiy for fruitful discussions. This project has been supported by the Russian Fund for Basic Research grant 12-01-00450-a and by the EMC student grant. Work of the second author has also been supported by the Russian Presidential Grant Programme for Young Ph.D.'s, grant no. MK-6628.2012.1, for Leading Scientific Schools, grant no. NSh-3229.2012.1, and RFBR grants 11-01-12135-ofi-m-2011 and 11-01-00760-a.

## References

1. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems **30**(1–7) (1998) 107–117
2. Granka, L.A., Joachims, T., Gay, G.: Eye-tracking analysis of user behavior in WWW search. In: Proceedings of the 27$^{th}$ Annual ACM SIGIR Conference. (2004) 478–479
3. Craswell, N., Zoeter, O., Taylor, M., Ramsey, B.: An experimental comparison of click position-bias models. In: Proceedings of the 1$^{st}$ ACM International Conference on Web Search and Data Mining. (2008) 87–94
4. Zhang, V., Jones, R.: Comparing click logs and editorial labels for training query rewriting. In: Query Log Analysis: Social And Technological Challenges, 16$^{th}$ WWW Conference workshop. (2007)
5. Dupret, G., Piwowarski, B.: A user browsing model to predict search engine click data from past observations. In: Proceedings of the 31$^{st}$ Annual ACM SIGIR Conference. (2008) 331–338
6. Chapelle, O., Zhang, Y.: A dynamic Bayesian network click model for web search ranking. In: Proceedings of the 18$^{th}$ International Conference on World Wide Web. (2009) 1–10
7. Zhang, Y., Chen, W., Wang, D., Yang, Q.: User-click modeling for understanding and predicting search-behavior. In: Proceedings of the 17$^{th}$ ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA (2011) 1388–1396
8. Guo, F., Liu, C., Kannan, A., Minka, T., Taylor, M., Wan, Y., Faloutsos, C.: Click chain model in web search. In: Proceedings of the 18$^{th}$ International Conference on World Wide Web. (2009) 11–20
9. Hu, B., Zhang, Y., Chen, W., Wang, G., Yang, Q.: Characterize search intent diversity into click models. In: Proceedings of the 20$^{th}$ International Conference on World Wide Web. (2011) 17–26
10. Srikant, R., Basu, S., Wang, N., Pregibon, D.: User browsing models: relevance versus examination. In: Proceedings of the 16$^{th}$ ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2010) 223–232
11. Zhu, Z., Chen, W., Minka, T., Zhu, C., Chen, C.: A novel click model and its applications to online advertising. In: Proceedings of the 3$^{rd}$ ACM International Conference on Web Search and Data Mining. (2010) 321–330
12. Zhang, Y., Wang, D., Wang, G., Chen, W., Zhang, Z., Hu, B., Zhang, L.: Learning click models via probit Bayesian inference. In: Proceedings of the 19$^{th}$ ACM International Conference on Information and Knowledge Management. (2010) 439–448

13. Dupret, G., Liao, C.: A model to estimate intrinsic document relevance from the clickthrough logs of a web search engine. In: Proceedings of the 3rd ACM International Conference on Web Search and Data Mining. (2010) 181–190
14. Yandex: Yandex Internet Mathematics competition. `http://imat-relpred.yandex.ru/` (2011)
15. Ling, C.X., Huang, J., Zhang, H.: Auc: a statistically consistent and more discriminating measure than accuracy. In: Proceedings of the International Joint Conference on Artificial Intelligence 2003. (2003) 519–526
16. Minka, T., Winn, J., Guiver, J., Knowles, D.: Infer.NET 2.4 (2010) Microsoft Research Cambridge. http://research.microsoft.com/infernet.
17. Zheng, Z., Chen, K., Sun, G., Zha, H.: A regression framework for learning ranking functions using relative relevance judgments. In: Proceedings of the 30th Annual ACM SIGIR Conference, ACM (2007) 287–294
18. Hastie, T., Tibshirani, R., Friedman, J.: Elements of Statistical Learning. Springer, New York (2008)
19. Donmez, P., Svore, K.M., Burges, C.J.C.: On the local optimality of lambdarank. In: Proceedings of the 32nd Annual ACM SIGIR Conference, ACM (2009) 460–467
20. Burges, C.J.C.: From RankNet to LambdaRank to LambdaMART : An overview. Technical report, Microsoft Research (2010)
21. Burges, C.J.C., Svore, K.M., Bennett, P.N., Pastusiak, A., Wu, Q.: Learning to rank using an ensemble of lambda-gradient models. Journal of Machine Learning Research **14** (2011) 25–35
22. Friedman, J.: Greedy function approximation: a gradient boosting machine. Annals of Statistics **29** (2001) 1180
23. Hu, B., Liu, N.N., Chen, W.: Learning from click model and latent factor model for relevance prediction challenge. In: Proceedings of the Workshop on Web Search Click Data, WSDM 2012. (2012)
24. Figurnov, M., Kirillov, A.: Linear combination of random forests for the relevance prediction challenge. In: Proceedings of the Workshop on Web Search Click Data, WSDM 2012. (2012)