A Whole Page Click Model to Better Interpret Search Engine Click Data

Weizhu Chen[1,2] and **Zhanglong Ji**[1]
wzchen, v-zhaj@microsoft.com
Microsoft Research Asia[1]

Si Shen[2] and **Qiang Yang**[2]
wzchen, shensi, qyang@cse.ust.hk
Hong Kong Univ. of Science & Technology[2]

Abstract

Recent advances in click modeling have established it as an attractive approach to interpret search click data. These advances characterize users' search behavior either in advertisement blocks, or within an organic search block through probabilistic models. Yet, when searching for information on a search result page, one is often interacting with the search engine via an entire page instead of a single block. Consequently, previous works that exclusively modeled user behavior in a single block may sacrifice much useful user behavior information embedded in other blocks. To solve this problem, in this paper, we put forward a novel Whole Page Click (WPC) Model to characterize user behavior in multiple blocks. Specifically, WPC uses a Markov chain to learn the user transition probabilities among different blocks in the whole page. To compare our model with the best alternatives in the Web-Search literature, we run a large-scale experiment on a real dataset and demonstrate the advantage of the WPC model in terms of both the whole page and each block in the page. Especially, we find that WPC can achieve significant gain in interpreting the advertisement data, despite of the sparsity of the advertisement click data.

Introduction

Click-through logs in a search engine are considered an invaluable resource containing user preference information on search result or online advertisements. Their analysis can benefit a wide range of search-related applications, such as Web search ranking (Yoshiyuki Inagaki 2010), ads click-through rate prediction (Richardson, Dominowska, and Ragno 2007), or user satisfaction estimation (Dupret and Liao 2010). In the process of analyzing the click-through logs, a central issue is how to infer the user-perceived relevance for each query-document pair. With this relevance learnt from the massive search click data, a commercial search engine can understand search users better and provide more accurate search results. Recently, many attempts have been made on learning the user-perceived document relevance from the click-through data, in an effort to formalize it as a click modeling problem. Typical works include dynamic Bayesian network (DBN) model (Chapelle and Zhang 2009), user browsing model (UBM) (Dupret and Piwowarski 2008), click chain model (CCM) (Guo et al. 2009), and session utility model (SUM) (Dupret and Liao 2010).

Although click data are informative, a well-known challenge for click modeling is position bias, whereby a document in higher position is likely to attract more user clicks even though it may be not as relevant as documents in lower positions. Thus, the widely accepted metric of using click-through rate (CTR) cannot be used as a precise measurement for document relevance. This bias was firstly noticed by (Granka, Joachims, and Gay 2004) in their eye-tracking experiment. Thereafter, (Richardson, Dominowska, and Ragno 2007) studied the advertisement problem and proposed to increase the relevance of documents in lower position by a multiplicative factor. (Craswell et al. 2008) formalized this idea as the examination hypothesis to model organic search problem. Most of the subsequent works follow the examination hypothesis to interpret user click behavior.

In spite of their successes, previous works on click modeling assume that a user only examines the results in a single block, as shown in Figure 1. Typical models like DBN and UBM characterize user behavior only in organic search blocks, while joint relevance examination (JRE) model (Srikant et al. 2010) and general click model (GCM) (Zhu et al. 2010) interpret user click behavior exclusively in ads blocks. In particular, they simply ignore user click behavior in other blocks. However, it is obvious that carrying out this simplification might sacrifice useful information in other blocks. For example, a user might perform some clicks on the organic search block and feel disappointed, and then switch to make a click on the related searches block to change to another query. Without taking the click in the latter block into consideration, it may mislead some previous click models, such as DBN, to believe that one is satisfied with the last click in the first block, and thus one can infer an inaccurate relevance. Moreover, it can be observed from the logs (Table 1) that the user behavior of interacting with multiple blocks is common when users seek information in search engines. Yet few attempts have been made to interpret the click behavior in a whole page with multiple blocks, nor to explain the click relationship among different blocks.

Contributions. This line of thinking leads to the Whole Page Click (WPC) model that we present in this paper. We

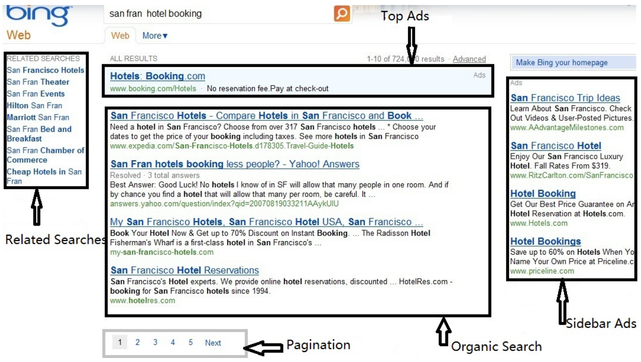

Figure 1: *Blocks in a whole page*

design the WPC model as a nested structure. The outer layer in this nested structure is to learn transition probabilities between blocks from the data, which is modeled via a Markov chain. In the inner layer, popular click models designed for a single block can be integrated into WPC. We use UBM as the inner model for explanation with a demonstration of the extension to other click models. Finally, We conduct extensive experiments on a real dataset from a commercial search engine and evaluate the WPC model in terms of interpreting search click data. We empirically show that WPC is capable of learning on large-scale data. The experiments are conducted with about 90 million sessions. Experimental results demonstrate that the WPC model can outperform previous works significantly. Especially, with the modeling of the whole page, we can leverage a huge amount of click data in organic search blocks to significantly boost the performance of prediction in the ads blocks, which is believed to be a challenging problem in click model literature.

In the rest of the paper, we first present preliminaries about click models and then introduce the WPC model. After that we discuss the experimental results and conclude our work.

Preliminaries

Before delving into modeling details, we introduce some definitions and background that are used throughout the paper. A user starts a *session* by issuing a *query* to a search engine, which returns a *Search Engine Result Page* (SERP) containing multiple blocks. We use $d_\phi(i)$ to indicate the document ranked at the position i in a specific block. The user then examines the SERP and clicks some or none of the documents. Any subsequent query re-submission or reformulation is regarded as initiating a new session.

In click models, examinations and clicks are treated as probabilistic events. For a particular session, we use a binary random variable $E_i = 1$ to indicate that the document at the position i is examined and otherwise $E_i = 0$. Similarly, we use $C_i = 1$ to indicate the document at the position i in organic search block is clicked and otherwise $C_i = 0$. Therefore, $P(E_i = 1)$ indicates the examination probability for document at position i and $P(C_i = 1)$ is the corresponding click probability.

Examination Hypothesis

The *examination hypothesis* assumes that a displayed document is clicked if and only if this document is both *examined* and perceived to be *relevant*. The relevance of a document is a query-specific variable which directly measures the probability that a user will click on document given that it has been examined by the user. More precisely, given a query q and a document $d_{\phi(i)}$ at position i, the examination hypothesis assumes the probability of the binary click event C_i as follows:

$$P(C_i = 1|E_i = 0) = 0 \tag{1}$$

$$P(C_i = 1|E_i = 1, q, d_{\phi(i)}) = a_{\phi(i)} \tag{2}$$

where $a_{\phi(i)}$ measures the degree of relevance between the query q and the document $d_\phi(i)$. Obviously, $a_{\phi(i)}$ is the conditional probability of a click after examination. Thus, the Clickthrough Rate (CTR) is represented as

$$P(C_i = 1) = \underbrace{P(E_i = 1)}_{position\ bias} \underbrace{P(C_i = 1|E_i = 1)}_{document\ relevance} \tag{3}$$

where CTR is decomposed into position bias and document relevance.

Following the examination hypothesis, given the condition E_i, the relevance of the document is a constant value. However, a challenge in this decomposition is that whether a document is examined or not is not observable from clickthrough logs, so subsequent click models try to formalize this examination event as a hidden variable and make different assumptions to deduce its probability.

An important extension of the examination hypothesis is the user browsing model (UBM). It assumes that the examination event E_i depends not only on the position i but also on previous clicked position l_i in the same session, where $l_i = \max\{j \in \{1, \cdots, i-1\} \mid C_j = 1\}$. It introduces a series of global parameters $\beta_{l_i,i}$ to measure the transition probability from position l_i to position i. Formally, the UBM is characterized by the following equations:

$$P(E_i = 1|C_{1:i-1} = 0) = \beta_{0,i} \tag{4}$$

$$P(E_i = 1|C_{l_i} = 1, C_{l_i+1:i-1} = 0) = \beta_{l_i,i} \tag{5}$$

$$P(C_i = 1|E_i = 0) = 0 \tag{6}$$

$$P(C_i = 1|E_i = 1) = a_{\phi(i)} \tag{7}$$

Here $l_i = 0$ if there are no preceding clicks. The term $C_{i:j} = 0$ is an abbreviation for $C_i = C_{i+1} = \cdots = C_j = 0$.

Cascade Model

The *cascade model* assumes that users always examine documents from top to bottom without skipping. Therefore, a document is examined only if all previous documents are examined. For an examined document, whether it is clicked or not still depends on its relevance. But the click events depend on the relevance of all the documents shown above. Formally, the cascade model can be formalized as following assumptions:

$$P(E_1 = 1) = 1 \tag{8}$$

$$P(E_{i+1} = 1|E_i = 0) = 0 \tag{9}$$

$$P(C_i = 1|E_i = 1) = a_{\phi(i)} \tag{10}$$

$$P(E_{i+1} = 1|E_i = 1, C_i) = 1 - C_i \tag{11}$$

The fourth assumption above implies that a user abandons the session if she gets a desired search result; otherwise she always continues the examination. This simultaneously reveals that it can only be applied to the sessions with at most one click. Yet this is too strict for real logs with multiple clicks in a session. Thus several models are introduced to deal with multiple clicks within a session based on cascade model.

The *dynamic Bayesian network* (DBN) model (Chapelle and Zhang 2009) emphasizes that a click does not necessarily indicate a user's satisfaction with the document. Instead, a user may be attracted by some misleading titles or snippets to trigger a click. Therefore, the DBN model distinguishes document relevance as *perceived relevance* a_i and *actual relevance* s_i. Whether a user clicks on a document or not depends on its perceived relevance, and whether the user is satisfied with the document depends on the actual relevance. If the user is satisfied with the clicked document, she does not examine the next document. Otherwise, there is a probability $1-\gamma$ that the user abandons her session and a probability γ that she continues her search. Thus, the DBN model replaces the last assumption in the cascade model by the following transition equations:

$$P(S_i = 1|C_i = 0) = 0 \quad (12)$$

$$P(S_i = 1|C_i = 1) = s_{\phi(i)} \quad (13)$$

$$P(E_{i+1} = 1|S_i = 1) = 0 \quad (14)$$

$$P(E_{i+1} = 1|E_i = 1, S_i = 0) = \gamma \quad (15)$$

where S_i is a hidden event indicating if a user is satisfied with the document $d_{\phi(i)}$. The values of a_i and s_i are estimated by the Expectation-Maximization algorithm in the original paper, and there is a probit approach to infer the model introduced in (Zhang et al. 2010).

Model

In this section, we first introduce the statistics for the whole page user behavior and let it speak for the motivation. Then we present the notations to formalize the whole page user behavior and provide the model specifications.

Previous models focus on a single block in the SERP. However, after a user issues a query to a search engine, in reality the search engine returns a whole page containing multiple blocks. This whole page is composed of five blocks in general, as shown in Figure 1. To understand the density distribution of user behavior in each block, we process one week of log data in a popular search engine and list the click distribution in Table 1. We can observe from the table that although organic search block occupies a high percentage of the clicks (79.2%), the clicks in other blocks can not be ignored (16.7%). In particular, the clicks in the advertisement (about 6.7%) are vital since they contribute to the revenue for commercial search engines; and there are many specific works focusing on interpreting user click behavior in the ads blocks, such as JRE and GCM. On the flip side, we notice that about 16.2% of sessions have clicks in more than one blocks, which demonstrates that it is common for users to interact with search engines on a whole page scale and perform clicks in more than one block when they are using search engines.

Block Name	# Percentage
Organic Search	79.2
Top Ads	5.2
Sidebar Ads	1.5
Related Searches	5.7
Pagination	4.3
Others	4.1

Table 1: *Click distribution over different blocks*

Let us assume that a user examines a series of blocks when she interacts with the SERP. The sequence of these blocks in session s can be regarded as a route, denoted as R_s. We use N_s to denote the number of blocks in R_s and number them as a sequence $B_1, B_2, ..., B_{N_s}$. It should be noted that whether or not a block is examined by a user is not observable from logs, which therefore is modeled as a hidden variable in the inference section. For each block, we may classify it into two categories based on whether it has clicks in the logs within the session s. Thus, for the blocks with clicks in the session s, we denote the blocks as a sequence $B_{\pi(1)}, ..., B_{\pi(n_s)}$ where n_s is the number of clicked blocks and $\pi(i)$ gives the original numbering in R_s. We use $R_{\pi(s)}$ to represent this subsequence. Additionally, we use C to represent all the clicks in session s and C^i to represent the clicks in the i-th block. For any two consecutive clicked blocks, $B_{\pi(m)}$ and $B_{\pi(m+1)}$, where $m \in [1, n_s - 1]$, we use $R_{s,m}$ to denote all the blocks that a user examines between $B_{\pi(m)}$ and $B_{\pi(m+1)}$ excluding $B_{\pi(m+1)}$; thus $R_{s,m} = \{B_{\pi(m)}, ..., B_{\pi(m+1)-1}\}$. Accordingly, we use $C_{s,m}$ to denote the clicks within $R_{s,m}$. Clearly, $R_{s,m}$ may contain zero or more blocks that the user examines without a click, but this is not observable from the log data. This poses a challenge in designing a click model for a whole page.

In response to this issue, we design the WPC model as a two-layer structure. We call the two layers macro and micro models, respectively.. The macro model characterizes the user block switch behavior, i.e., user transition behavior among blocks. The micro model focuses on the user behavior inside a single block, such as user click or skip behavior of a search result within the organic search block.

When we process one session s in the logs with consideration of the whole page, we may observe all the clicked blocks $B_{\pi(1)}, ..., B_{\pi(n_s)}$ and all the clicks C in these blocks. Thus, we may have the following likelihood function:

$$P(R_{\pi(s)}, C) = \sum_{R_s \supset R_{\pi(s)}} P(C|R_s)P(R_s) \quad (16)$$

The goal of the following assumptions and inference is to maximize this likelihood. The term $P(R_s)$ represents the transition probability among different blocks in the path R_s. But neither the number nor the order of blocks in this path is observable. To address this problem, we first assume that the block transition in a route R_s is a Markov chain of order

N. With an example of $N = 1$, we may calculate $P(R_s)$ as:

$$P(R_s) = \prod_{i=0}^{N_s} P_t(B_{i+1}|B_i, C^i) = \prod_{m=0}^{n} P(R_{s,m}) \quad (17)$$

$$P(R_{s,m}) = \prod_{k=\pi(m)}^{\pi(m+1)-1} P_t(B_k|B_{k-1}, C^{k-1}) \quad (18)$$

Here we use P_t to denote the transition probability. The condition C^i separates the transition probability into two cases according to whether or not there is a click in block B_i. We observe that if there is a click in a given block, a user is less likely to examine next block. Moreover, we can use a padding technique to generalize this equation even further. We use B_0 as the initial status before one examines the first block. Thus, $P_t(B_1|B_0)$ is the probability that one starts the examination from block B_1. Similarly, we use B_{N_s+1} to represent the event that one leaves session s, and thus $P(B_{N_s+1}|B_{N_s})$ is the probability that one leaves the session right after examining block B_{s,N_s}. As a result, from the definition of $R_{s,m}$ and Equation 20, it is easy to see that Equation 19 holds.

Additionally, we make the assumption that each click solely depends on its current block, i.e., it is not influenced by other blocks. Thus, for term $P(C|R_s)$ in Equation 18, we can formalize it as follow

$$P(C|R_s) = \prod_{i=0}^{N_s} P(C^i|B_i) = \prod_{m=0}^{n} P(C_{s,m}|R_{s,m}) \quad (19)$$

Here the micro model $P(C^i|B_i)$ only depends on a single block B_i, thus it can be formalized by general click models introduced in section *Preliminaries* . Take *UBM* as an example for a micro model. $P(C^i|B_i)$ is the product among β, α and $(1 - \beta \cdot \alpha)$.

Inference

To calculate the parameters θ in WPC, both in the macro model and in the micro model, we need to infer the corresponding parameter updating equations. Here we use the *UBM* model as an example for micro model and illustrate the inference in details. The macro model has parameters $P_t(B_i|B_j)$ and the *UBM* micro model has parameters α, β, thus $\theta = \{P_t(B_i|B_j), \alpha, \beta\}$.

To estimate these parameters from the logs, we use the Expectation-Maximization(*EM*) algorithm to maximize the likelihood of Equation 18. Based on Equation 19 and 21, we can compute the complete likelihood in EM, where $R_{s,m}$ is regarded as hidden variables. Following the general EM algorithm in (Dempster, Laird, and Rubin 1977), we get the expectation of the complete likelihood evaluated for θ as:

$$Q(\theta|\theta^t) = \quad (20)$$
$$\sum_{s} \sum_{m=0}^{n} \sum_{R_{s,m}} P(R_{s,m}|C_{s,m}, \theta^t) \cdot log(P(R_{s,m}, C_{s,m}|\theta))$$

In the E step, we compute the posterior distribution of $P(R_{s,m}|C_{s,m}, \theta^t)$. However, computing this posterior may involve infinite number of $R_{s,m}$ since the block examination event is unobservable from the data. To tackle this problem, we make an assumption that each block can only appear at most once for each possible route of $R_{s,m}$, that is, the examination path between clicks cannot involve cycles. Since we only consider five blocks in the WPC model, the number of possible $R_{s,m}$ is limited. When $R_{s,m}$ contains two consecutive clicks, it is easy to deduce that there are only 16 possible routes for $R_{s,m}$. For the extreme case that $R_{s,m}$ contains only one click, which is generated by the padding technique, there are 64 possible routes for $R_{s,m}$. With this simplification, we may use the current parameter θ^t and the observable $C_{s,m}$ in each block to calculate the probability of each possible path $P(R_{s,m}|C_{s,m}, \theta^t)$. Thus, the probability of each possible $P(R_{s,m}|C_{s,m}, \theta^t)$ when processing a session s can be updated as follows.

Query Freq.	# Query	# Document	# Session
$10^0 \sim 10^1$	413,341	4,167,798	2,014,965
$10^1 \sim 10^{1.5}$	568,315	3,635,808	2,992,680
$10^{1.5} \sim 10^2$	723,175	10,314,368	10,115,500
$10^2 \sim 10^{2.5}$	653,848	10,281,005	30,809,145
$10^{2.5} \sim 10^3$	301,678	5,839,170	42,232,965
Total	2,660,357	32,238,149	88,165,255

Table 2: *A summary of the data set*

$$P(R_{s,m}|C_{s,m}, \theta^t) = \frac{P(R_{s,m}, C_{s,m}|\theta^t)}{\sum_{R_{s,m}} P(R_{s,m}, C_{s,m}|\theta^t)} \quad (21)$$

$$P(R_{s,m}, C_{s,m}|\theta^t) = P(C_{s,m}|R_{s,m}, \theta^t) P(R_{s,m}|\theta^t) \quad (22)$$

In this procedure, the E step completes an iteration after processing all sessions and then switch to the parameter updating in M step.

In the *M* step, we determine the revised parameter θ^{t+1} by maximizing this function:

$$\theta^{t+1} = \arg\max_{\theta} Q(\theta, \theta^t) \quad (23)$$

In this paper, we use the *UBM* model as an example for micro model and provide the updating equations in the Appendix. For other micro models, as mentioned above, it is similar to infer the updating equations by considering specific $P(C_{s,m}|B_i)$. After the M step, the EM algorithm uses the revised parameters to continue the E step in the next iteration. It runs for N iterations until all the parameters converge. For the WPC model, we use $N = 10$ and find that the model converges under all settings of parameters θ.

Experiments & Discussion

In this section, we conduct experiments to verify the effectiveness of the proposed model by comparing it with two click models, the DBN model and the UBM model.

Experiment Setups

Click logs data: The session data used to train and evaluate the click models are collected from a commercial search engine in the U.S. market in English over one week in Oct, 2010. To verify the effect of multiple blocks, we first filter

Model	Whole Page	Organic Search	Top Ads	Side Ads	Related Search	Pagination
DBN	1.088	1.175	1.530	1.084	1.0245	1.010
UBM	1.082	1.163	1.440	1.064	1.0238	1.009
WPC	1.067	1.156	1.213	1.019	1.0211	1.006
Improvement Over DBN	23.8%	10.8%	59.1%	77.0%	13.8%	0.4%
Improvement Over UBM	18.2%	4.3%	51.5%	70.3%	11.3%	0.33%

Table 3: *Perplexity in five blocks for three click models.*

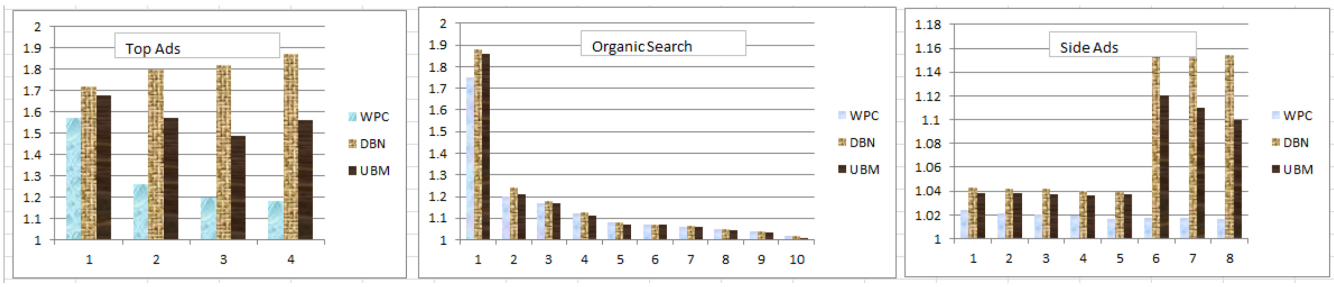

Figure 2: *Perplexity over different positions*

queries whose SERP contain less than three blocks. Moreover, in order to prevent the whole dataset being dominated by the extremely frequent queries, we limit the number of sessions for each query to be the first 10^3. We then divide the whole dataset into 5 subsets based on the frequency of a query, and split each subset into training set and test set evenly. In total, we collect 2,660,357 distinct queries and 88,165,255 sessions. The detailed information is summarized in Table 2.

After a click model estimates its parameters in training, we compare the observed and predicted CTR. The prediction accuracy is evaluated by click perplexity, which has been widely used to measure the goodness-of-fit of click models, such as GCM, UBM and BBM. A smaller perplexity indicates a better prediction accuracy and the optimal value is one. For a given position i and a set of sessions $s_1, s_2, \cdots, s_n$, we use $c_1, c_2, \cdots, c_n$ to denote the binary click events of the i-th document in each session. Let $q_1, q_2, \cdots, q_n$ denote the predicted click probability by the click model. The perplexity p_i for the position i is:

$$p_i = 2^{\frac{1}{n}\sum_{i=1}^{n}(c_i \log_2 q_i + (1-c_i)\log_2 q_i)} \quad (24)$$

The perplexity of the entire dataset is averaged over all positions, and the improvement of perplexity value p_a over p_b is calculated as $(p_b - p_a)/(p_b - 1) \times 100\%$ (Guo et al. 2009).

Discussion

In Table 3, we report the perplexity of DBN, UBM and WPC on the test dataset in two settings: over a whole page and over each individual block on the page.. For the whole page, we consider all the blocks together when calculating the perplexity. It is clear to see that WPC can achieve 23.8% and 18.2% relative improvements over DBN and UBM, respectively. This is consistent with the work of (Zhang et al. 2010) in that UBM is slightly better than DBN in terms of perplexity. We perform the significance test for these improvements, and find that the p-value of t-test are both less than 0.01% due to the large-scale dataset. These improvements verify the necessity of modeling user click behavior in a whole page to better interpret search click data.

If we look at the third column in Table 3 for the results of organic search block, we find that the improvements (4.3% over UBM and 10.8% over DBN) are not as significant as that in the whole page. This may be attributed to the fact that organic search block data occupy a large percentage of the click, as shown in Table 1. For a huge number of queries, their click data are sufficient for the single block click models, like UBM, to attain a good accuracy. Yet, even though the improvements in this block are limited, WPC still outperforms existing models.

More can be observedd from the improvements in the fourth and fifth columns of Table 3 for both ads blocks. The improvements over DBN and UBM are 59.1% and 51.5% respectively in the top ads block. Compared with the organic search data, the biggest difference in advertisement data is the sparsity, since few search users are likely to click on advertisements. This might indicate that existing click models are not effective to interpret the advertisement data due to its sparsity. However, by characterizing the information in other blocks with larger amount of click information, the whole page click model can significantly improve the interpretation of user click behavior in ads blocks. It is not surprising to see that the improvements on the sidebar ads block (77.0% over DBN and 70.3% over UBM) are more significant since the click behavior in this block is more sparse.

To draw a conclusion about the consistency of the improvements, we analyze the improvements for above three blocks in terms of different positions. The experimental results for top ads block, organic search block and sidebar ads block are reported in Figure 2 respectively. It illustrates that WPC can achieve a better or comparable perplexity in almost all positions in all blocks. The exception happens in lower positions, like position 9 or 10, within the organic search block, where WPC is slightly worse than UBM. This may be attributed to the fact that in lower positions of organic search block, the existing information in the current block, like the click or skip information in above positions, is already indicative enough to characterize the user click behavior. This leads to an intuitive conclusion that providing information in other blocks is not always beneficial when the click information in the current block is good enough. However, the improvements on the ads blocks are consistent and very significant in all positions. This provides us a clear confirmation that WPC can bring significant improvements for interpreting advertisement data.

Conclusion and Extensions

In this paper, we have investigated the necessity of characterizing search click data via a whole page, instead of a single block. We have proposed a WPC model that can characterize user click behavior in multiple blocks and demonstrated that it can perform better than the DBN and UBM models. Especially, we have demonstrated that WPC can bring a major improvement of interpreting advertisement data and verified our findings by large-scale experiments on a real dataset.

Modeling user behavior in a whole page can be used for re-ranking or training a better ranker in the future. one extension is to analyze user behavior in different blocks and verify whether the position bias varies in different blocks. Another extension is to use the block switching behavior to learn the user satisfaction in the whole page. Both are topics for future work.

Acknowledgements

We thank the support of the Microsoft Research Asia grant MRA10EG01 Hong Kong RGC grant 621010 and RGC/NSFC Joint research grant N_HKUST 624/09.

References

Chapelle, O., and Zhang, Y. 2009. A dynamic bayesian network click model for web search ranking. WWW '09, 1–10.

Craswell, N.; Zoeter, O.; Taylor, M.; and Ramsey, B. 2008. An experimental comparison of click position-bias models. ACM WSDM '08, 87–94.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B* 39(1):1–38.

Dupret, G., and Liao, C. 2010. A model to estimate intrinsic document relevance from the clickthrough logs of a web search engine. ACM WSDM '10, 181–190.

Dupret, G. E., and Piwowarski, B. 2008. A user browsing model to predict search engine click data from past observations. ACM SIGIR '08, 331–338.

Granka, L. A.; Joachims, T.; and Gay, G. 2004. Eye-tracking analysis of user behavior in www search. ACM SIGIR '04, 478–479.

Guo, F.; Liu, C.; Kannan, A.; Minka, T.; Taylor, M.; Wang, Y.-M.; and Faloutsos, C. 2009. Click chain model in web search. WWW '09, 11–20.

Richardson, M.; Dominowska, E.; and Ragno, R. 2007. Predicting clicks: estimating the click-through rate for new ads. WWW '07, 521–530.

Srikant, R.; Basu, S.; Wang, N.; and Pregibon, D. 2010. User browsing models: Relevance versus examination. KDD '10, 223–232.

Yoshiyuki Inagaki, Narayanan Sadagopan, G. D. C. L. A. D. Y. C. Z. Z. 2010. Session based click features for recency ranking. AAAI '10, 1334–1339.

Zhang, Y.; Wang, D.; Wang, G.; Chen, W.; Zhang, Z.; Hu, B.; and Zhang, L. 2010. Learning click models via probit bayesian inference. CIKM '10, 439–448.

Zhu, Z. A.; Chen, W.; Minka, T.; Zhu, C.; and Chen, Z. 2010. A novel click model and its applications to online advertising. ACM WSDM '10, 321–330.

Appendix

Here we provide the detailed deduction of WPC using UBM as the micro model. Following the UBM paper, we denote the indices of α_l, β_k and $(1-\beta_k\alpha_l)$ in the likelihood of $P(C_{s,m}|R_{s,m})$ as $S_{R_{s,m},l}$, $S_{R_{s,m},k}$ and $S_{R_{s,m},k,l}$. Similarly, the index of $P(B_j|B_i,c)$ is $S_{R_{s,m},i,j,c}$. We also define:

$$S_l = \sum_s \sum_{m=0}^{n_s} \sum_{R_{s,m}} \tilde{P}(R_{s,m}|C_{s,m},\theta^t) S_{R_{s,m},l}$$

$$S_k = \sum_s \sum_{m=0}^{n_s} \sum_{R_{s,m}} \tilde{P}(R_{s,m}|C_{s,m},\theta^t) S_{R_{s,m},k}$$

$$S_{k,l} = \sum_s \sum_{m=0}^{n_s} \sum_{R_{s,m}} \tilde{P}(R_{s,m}|C_{s,m},\theta^t) S_{R_{s,m},k,l}$$

$$S_{i,j,c} = \sum_s \sum_{m=0}^{n_s} \sum_{R_{s,m}} \tilde{P}(R_{s,m}|C_{s,m},\theta^t) S_{R_{s,m},i,j,c}$$

After that we have the explicit expression of $Q^{t+1}(\theta|\theta^t)$:

$$Q^{t+1}(\theta|\theta^t) = \sum_l S_l \log(\alpha_l) + \sum_k S_k \log(\beta_k)$$
$$+ \sum_{k,l} S_{k,l} \log(1-\beta_k\alpha_l) + \sum_{i,j,c} S_{i,j,c} \log(P(B_j|B_i,c))$$
$$s.t. \sum_{j\neq i} P(B_j|B_i,c) = 1 (\forall i,c)$$

Thus, by maximization $Q^{t+1}(\theta|\theta^t)$, the final parameter updating equations are:

$$\alpha_l^{t+1} = \frac{1}{S_l + \sum_k S_{k,l}} \left(S_l + \sum_k \frac{S_{k,l}\alpha_l^t(1-\beta_k^t)}{1-\beta_k^t\alpha_l^t} \right)$$

$$\beta_k^{t+1} = \frac{1}{S_k + \sum_l S_{k,l}} \left(S_k + \sum_l \frac{S_{k,l}\beta_k^t(1-\alpha_l^t)}{1-\beta_k^t\alpha_l^t} \right)$$

$$P_t^{t+1}(B_j|B_i,c) = \frac{S_{i,j,c}}{\sum_{j\neq i} S_{i,j,c}}$$