User-click Modeling for Understanding and Predicting Search-behavior

Yuchen Zhang[1], Weizhu Chen[1,2], Dong Wang[1], Qiang Yang[2]
Microsoft Research Asia, Beijing, China[1]
Hong Kong University of Science and Technology, Hong Kong[2]
{v-yuczha, wzchen, v-dongmw}@microsoft.com, {wzchen, qyang}@cse.ust.hk

ABSTRACT

Recent advances in search users' click modeling consider both users' search queries and click/skip behavior on documents to infer the user's perceived relevance. Most of these models, including dynamic Bayesian networks (DBN) and user browsing models (UBM), use probabilistic models to understand user click behavior based on individual queries. The user behavior is more complex when her actions to satisfy her information needs form a search session, which may include multiple queries and subsequent click behaviors on various items on search result pages. Previous research is limited to treating each query within a search session in isolation, without paying attention to their dynamic interactions with other queries in a search session.

Investigating this problem, we consider the sequence of queries and their clicks in a search session as a task and propose a task-centric click model (TCM). TCM characterizes user behavior related to a task as a collective whole. Specifically, we identify and consider two new biases in TCM as the basis for user modeling. The first indicates that users tend to express their information needs incrementally in a task, and thus perform more clicks as their needs become clearer. The other illustrates that users tend to click fresh documents that are not included in the results of previous queries. Using these biases, TCM is more accurately able to capture user search behavior. Extensive experimental results demonstrate that by considering all the task information collectively, TCM can better interpret user click behavior and achieve significant improvements in terms of ranking metrics of NDCG and perplexity.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]:

General Terms

Algorithms, Experimentation, Performance

KDD'11, August 21–24, 2011, San Diego, California, USA.

Keywords

Click Log Analysis, Task-Centric Click Model

1. INTRODUCTION

Search engine click-through logs are an invaluable resource that can provide a rich source of data on user preferences in their search results. The analysis of click-through logs can be used in many search-related applications, such as web search ranking [1], predicting click-through rate (CTR) [16], or predicting user satisfaction [7]. In analyzing click-through logs, a central question is how to construct a click model to infer a user's perceived relevance for each query-document pair based on a massive amount of search click data. Using a click model, a commercial search engine can develop a better understanding of search users' behavior and provide improved user services. Previous investigations of click models include dynamic Bayesian networks (DBN) [4], the user browsing model (UBM) [8], the click chain model (C-CM) [11] and the pure relevance model (PRM) [18].

While previous research seeks to model a user's click behavior based on browsing and click actions after she enters a single query, often several queries are entered sequentially and multiple search results obtained from different queries are clicked to accomplish a single search task. Take for example a typical scenario. A user may first issue a query, examine the returned results and then click on some of them. If the existing results do not satisfy her information needs, she may narrow her search and reformulate her query to construct a new query. This process can be repeated until she finds the desired results or gives up. Clearly, a typical search can include complex user behavior, including multiple queries and multiple clicks for each query, etc. Collectively, all the user's actions provide an overall picture of the user's intention as she interacts with the search engine. The multiple queries, clicked results, and underlying documents are all sources of information that can help reveal the user's search intent.

Traditionally, user sessions are obtained from a consecutive sequence of user search and browsing actions within a fixed time interval [15]. These sessions can be partitioned into two categories: a (query) session and a search session, where the former refers to the browsing actions for an individual query while the latter encompasses all queries and browsing actions that a user performs to satisfy her information need. In this paper, we consider the latter, and refer to a search session as a *task*. As mentioned above, the previous research considers query sessions only, but ignores other sources of information and their relations to the same task.

Thus, most previous research suffers from a lack of accuracy in many cases. The DBN model, for example, assumes that users are always satisfied with the last click of each query, without considering subsequent queries and clicks.

Contributions.

The above line of thinking has led us to consider the advantage of a task-centric click model (TCM) in this paper for understanding and predicting click behavior. In this paper, we first point out the necessity of modeling task level user behavior by letting the real data speak for itself. We then define and describe two new user-biases that influence a search task but have been ignored in previous investigations. We address these biases via our TCM.

The first bias indicates that users tend to express their information needs incrementally and then perform more clicks as their needs become clearer. The second bias illustrates that users tend to click on fresh documents that they have not seen before under the same task. We design our TCM using a probabilistic Bayesian method to address these two biases. TCM is general enough to integrate most other existing click models. Finally, we verify the effectiveness of the TCM by comparing its performance to the DBN and UBM models. We conduct experiments with a large-scale real-world dataset which shows that the TCM can be scaled up. Our experiments used more than 9.5 million search tasks as the research dataset. The experimental results show that, by considering all of the task information, the TCM can better model and interpret user click behavior and achieve significant improvements in terms of NDCG and perplexity.

2. PRELIMINARIES & RELATED WORKS

We start by introducing some background concerning traditional click models and related works for mining search session information for search-related applications.

2.1 Click Models

A well-known challenge for click modeling is the position bias. This bias was first noticed by Granka et al. [10], which states that a document appearing in a higher position is likely to attract more user clicks even if it is not relevant. Thereafter, Richardson et al. [16] proposed to increase the relevance of documents in lower positions by a multiplicative factor; Craswell et al. [6] later formalized this idea as an examination hypothesis. Given a query q and a document $d_{\phi(i)}$ at the position i, the examination hypothesis assumes the probability of the binary click event C_i given the examination event E_i as follows:

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

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

Here we use $C_i = 1$ to indicate the document at the position i is clicked and otherwise $C_i = 0$, with a similar definition for E_i. Moreover, $a_{\phi(i)}$ 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 Click-Through 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.

One important extension of the examination hypothesis is the UBM. It assumes that the examination event E_i depends not only on the position i but also on the previous clicked position l_i in the same query session, where $l_i = \max\{j \in \{1, \cdots, i-1\} | C_j = 1\}$, and $l_i = 0$ means no preceding clicks. Global parameters $\beta_{l_i,i}$ measure the transition probability from position l_i to position i, and $C_{i:j} = 0$ is an abbreviation for $C_i = C_{i+1} = \cdots = C_j = 0$:

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

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

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

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

A similar investigation using UBM is the Bayesian browsing model (BBM) [14], which adopts a Bayesian approach for inference with each random variable as a probability distribution. This is similar to the work on the General Click Model (GCM)[22]. It extends the model to consider multiple biases and shows that previous models are special cases of GCM. Hu et al. [12] extend UBM to characterize the diversity of search intents in click-through logs. Chen et al. [5] proposed a whole-page click model which considers the search result page including the organic search and advertising entries as a whole to help the CTR prediction.

Another extension is the cascade model. It assumes that users always examine documents without skipping from top to bottom. Therefore, a document is examined only if all previous documents are examined.

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

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

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

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

Two important improvements to the cascade model are the CCM [11] and DBN [4] models. Both emphasize that the examination probability also depends on the clicks and the relevance of previous documents. Moreover, allow users to stop the examination. CCM uses the relevance of previous documents for this while DBN uses a satisfaction parameter s_i. The parameter states that if the user is satisfied with the clicked document, she will not examine the next document. Otherwise, there is a probability γ that the user will continue her search.

$$P(S_i = 1|C_i = 0) = 0 \tag{12}$$

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

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

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

where S_i is a hidden event indicating user satisfaction.

There are three other models that do not employ the cascade assumption. The session utility model (SUM) [7], given a single query, measures the relevance of a set of clicked documents as the probability that a user stops the query session. The adPredictor model [9] interprets the click-through rate as a linear combination of weighted features. The pure relevance model (PRM) [18] states that the relevance of a document is not a constant but affected by clicks in other positions.

The research presented in this paper differs in its assumptions and approach from the previous research summarized above. We focused on how to explore the whole search session as an integrated and dynamic entity including multiple queries and query sessions. It incorporates the data from the

Table 1: The click rate on query sessions with respect to the position of the query session in the task. It is observed that users tend to click more on the last query session in a task.

# of Sessions in Task	First Session	Second Session	Third Session	Fourth Session	Fifth Session
1-Session Task	62.9%	-	-	-	-
2-Session Task	46.7%	65.7%	-	-	-
3-Session Task	48.4%	49.9%	67.0%	-	-
4-Session Task	47.8%	50.1%	49.2%	65.5%	-
5-Session Task	47.5%	48.6%	48.4%	49.5%	65.3%

whole search session to develop a more nuanced and effective click model.

2.2 Search Session Mining

Search session information has been used for many search applications. The single query is often ambiguous through and hard to use as an accurate representation of a user's intent. Thus, several works use previous queries or click behavior within the same search session to enrich the current query. White et al. [19] represented the search session information as ODP categories and used them to predict user interests. Xiang et al. [20] considered how users reformulate queries and used this information for Web search ranking. Shen et al. [17] proposed a method for context-aware ranking by enriching the current query with search session information. Cao et al. used conditioned random field and a hidden Markov model to model search session information for query classification [2] and query suggestion [3]. Our work differs from these studies. We focus on a click modeling problem: how to understand and predict user click behavior by learning the user's perceived relevance for each query-document pair. Our proposed model is a generative model, which learns its parameters by maximizing the whole search session likelihood with considering previous click model assumptions.

3. TASK-CENTRIC BEHAVIOR ANALYSIS

When a user is searching information in a search engine, however, she is performing a search task instead of a single query session, where a task may contain one or multiple query sessions. Simultaneously, user behavior in different query sessions under the same task should not be treated as the same or independent. There might exist some relationship between them. In this section, we process a real dataset to verify this assumption to obtain some findings to make the case for the necessity of task-centric modeling. For this motivating experiment, we collected the dataset in one week in September, 2010 from a commercial search engine. The dataset consists of 9.6 million tasks and 21.4 million query sessions. To have a better understanding of this dataset, we grouped all the search tasks by the number of query sessions in each task. More than 49.8% of the tasks contain more than one query session and include 77.5% of the search traffic. The proportion of tasks containing query session numbers from 2 to 5 are 21.0%, 10.7%, 6.4% and 4.1% respectively.

A well-known metric for characterizing user click behavior from search logs is the click-through rate (CTR). In the first designed experiment, we first explore variances in CTR (a query session is "clicked" if any of its documents is clicked) in terms of the position of each query session under a task. The results are presented in Table 1. It is observed that the CTR in the last query session, with a value ranging from 62.9% to 67.0%, is consistently higher despite the number of query sessions in a task. In other positions, the click rates are all significantly lower, ranging from 46.7% to 50.1%. This result clearly illustrates that users commonly tend to click more in the last query session of a task.

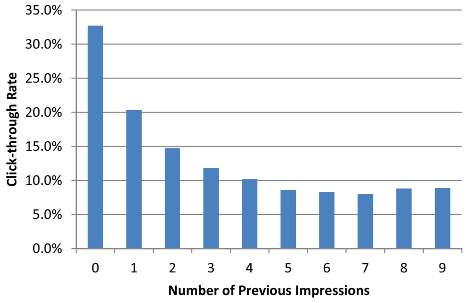

Figure 1: The probability of click on top documents with respect to the number of previous impressions.

In the second experiment, we focused on the users' behavior when the same document is presented to them more than once in the same task. To alleviate the effect of the position bias, we only consider the top position in this experiment. We found that in 23.6% of the tasks, the the top document appears more than once. In most cases this is because the repetitions of the same query, which happens when the user returns to the search engine after viewing one previous search result page and wants to check the remaining results. Even if the queries are different, the search engine may also return same documents because of the similarity of search intent. Moreover, what we are interested in is the user's click behavior when she sees a duplicate document. Thus we grouped the documents by the number of their previous presentations under the same task. Obviously, this number is 0 if the document is retrieved for the first time and larger than 0 for recurring documents. We report its CTR results in Figure 1. It shows that recurring documents have a significantly lower CTR. For documents which are presented for the first time, the averaged CTR is 32.7%. The CTR decreases to 20.3%, however, if there is one previous impression, and to 14.7% if there are two previous impressions. When a document has been presented five times, the CTR drops to 8.6% . A general observation is that users have a lower (but not zero) probability of clicking on stale results in the same task; i.e., users like to click fresh results.

4. TASK-CENTRIC CLICK MODEL

In this section, we propose a new model framework interpreting for the observations in Section 3. We have given two assumptions on the user's behavior in the tasks. Then, we designed the model according to the assumptions. In the next section, we will introduce a parameter estimation algorithm to infer the model.

4.1 Assumptions

Task (search session) identification is on-going work. There is no specific evidence to indicate that whether or not two adjacent query sessions belong to the same task. Thus we use the method proposed in work [15] with the default time threshold and similarity threshold to break the tasks. They have reported a promising result, and we assume it as a reliable method. Generally speaking, our observations reported in Section 3 involve two kinds of click biases:

1. Users tend to click more at the end of a task.
2. When a document is presented more than once, its CTR decreases after the first presentation. The more times it is presented under the same task, the less CTR it will have.

The first bias can be interpreted as follows. When a user is searching with an intent, especially for an informational intent or difficult intent, she might not know how to formalize a perfect query to represent it. In this case, she may first formalize a query to check its search results. Normally, the query may not reflect her final intent and the results may not be satisfactory. She may examine some results (snippets) without a click but learn from them to re-formalize her query. In this situation, there is no user click, but it does not indicate that all the results are irrelevant to the query. Instead, we may attribute the no click behavior to the mismatch between the users' intent and the results. Thus, before the final query, a user tends to investigate a perfect query to represent her intent. While in the last query, she is more likely to find the query matches her intent, and so she performs more clicks. This is consistent with what we observed in Table 1, where the CTR of the last query session is consistently high in a task. We formalize this assumption as follows:

Assumption 1 (Query Bias). *If a query does not match a user's intent, she will perform no clicks but learn from the search results to re-formalize a new query.*

It is worth noting that whether a query matches a user's intent or not cannot be observed from the logs. We need to model this as a random variable in the coming section. Besides the query bias mentioned above, the documents under different queries in the same task might have other relationships. Since all the queries under the same task are related, their returned documents may overlap each other. Thus, here we consider some documents which are presented in the same task more than once. When a document is examined the first time, the user will judge its usefulness and decide whether or not to click it. If a click happens, it means that she has acquired the information contained in the document. If no click happens, it means that the user is not interested in the document. In both cases, the document is less likely to be clicked again when it is repeatedly presented in the same task: i.e., users like fresh documents. This phenomenon, therefore, explains why the CTR decreases with the time of presentations in Figure 1. We formalize this assumption as:

Assumption 2 (Duplicate Bias). *When a document has been examined before, it will have a lower probability to be clicked when the user examines it again.*

4.2 Model

Based on the two assumptions mentioned above, we present our task-centric click model (TCM) in this section. The TCM has a two-layer structure. We call these layers the *macro model* and the *micro model*, respectively.

The macro model incorporates the query bias assumption into the TCM as illustrated in Figure 2. When a user submits a query to a search engine, the TCM first uses a random variable to characterize whether or not this query matches her intent. In both cases, the user starts browsing the search result page following the micro model introduced in the next paragraph. There is difference, however, between these two cases. If the query does not match the user's intent, there should be no click in the browsing process. Otherwise, the TCM will use a general micro model to interpret user click behavior and continue to model whether she will reformulate a new query in the same task. As is shown in figure 2, the TCM uses two binary random variables, M and N, to characterize the macro model. M represents whether the query is good, or in other words, whether it matches the user's intent, while N represents whether the user wants to continue her search if her previous query is good. It is worth noting that the value of M will influence how model interprets user behavior for the current query. The probability distribution, though, is not only learnt according to the information in the current query. Instead, it is a contextual variable which can be affected by the previous and following queries in the same task, i.e, it is learnt based on all of the task information. Intuitively, when all the task data are available, we use them to determine if the current query is appropriate to represent the user's intent for the whole task.

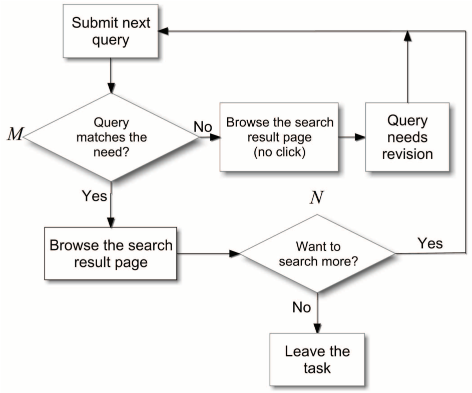

Figure 2: The Macro Model of TCM.

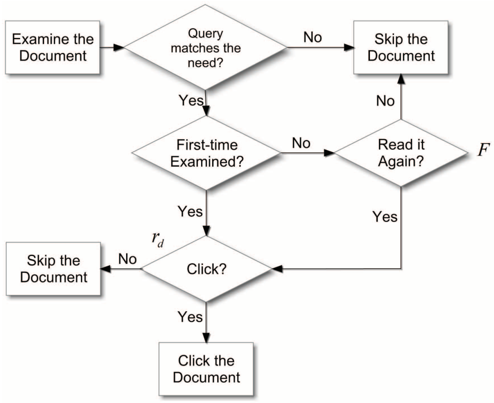

Figure 3: The Micro Model of TCM

The micro model, as shown in Figure 3, describes the browsing process in a single query session. We may integrate into the micro model existing click models while considering document freshness. Here in order to make the explanation easy to understand and the inference clear, we use a simple model to illustrate the definition of a micro model and illustrate its extension later to include other models. Our micro model is built with the examination hypothesis,

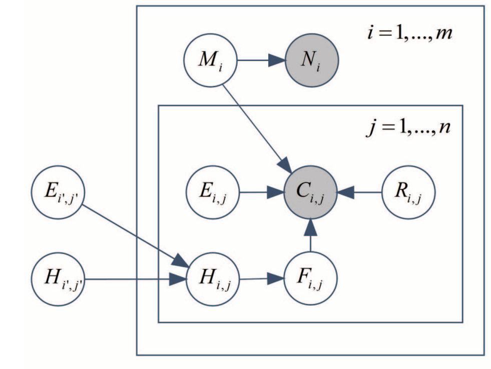

Figure 4: The graphical model of TCM.

which states that the user should first examine a document before clicking it. We assume the probability of a document being examined is uniquely determined by its ranking position. This strategy aims at eliminating the position bias in a query session. As we mentioned in the macro model, if a query does not reflect a user's intent, she will not click documents after the examination. Otherwise, it will turn to document freshness to judge whether the document has been examined in previous query sessions. If it is a first-time examination, we follow the general examination hypothesis, which states that the probability of a click is determined by the document's relevance, denoted as r_d. On the other hand, if the document has been examined before, she will decide whether the document is still "fresh", which is characterized as a random variable F. For a non-fresh document, its click behavior is no longer exclusively determined by its relevance, so we need to characterize the freshness to interpret user click behavior. In other words, the user will tend to skip a non-fresh document, since it might have been examined before. With the fact that $P(F = 1) \leq 1$, intuitively, we may regard the current design as multiplying a discount factor to the relevance of non-fresh documents. In this way, we follow the duplicate assumption in assumption 2 which states that the CTR of the document decreases after being examined more than once.

To give a formal definition to the TCM, we present it as a graphical model in Figure 4. In this graph, we assume that a task contains m queries (sessions), with each query session containing n ordered documents. We use (i, j) to indicate the j-th ranking position in the i-th query session. The symbols in Figure 4 and their descriptions are listed in Table 2. All the variables are binary variables.

Table 2: Notations Used in TCM

Symbol	Description
M_i	Whether the i-th query matches the user's intent.
N_i	Whether the user submits another query after the i-th query session.
$E_{i,j}$	Examination of the document at (i, j).
$H_{i,j}$	Previous Examination of the document at (i, j).
$F_{i,j}$	Freshness of the document at (i, j).
$R_{i,j}$	Relevance of the document at (i, j).
$C_{i,j}$	Whether the document at (i, j) is clicked.
(i', j')	Assume that d is the document at (i, j). i' is the latest query session where d has appeared in previous query sessions, and j' is the ranking position of this appearance.

To represent the TCM with another approach, we may formalize the definition in Figure 4 with the following conditional probabilities:

$$P(M_i = 1) = \alpha_1; \tag{16}$$
$$P(N_i = 1 | M_i = 1) = \alpha_2; \tag{17}$$
$$P(F_{i,j} = 1 | H_{i,j} = 1) = \alpha_3; \tag{18}$$
$$P(E_{i,j} = 1) = \beta_j \tag{19}$$
$$P(R_{i,j} = 1) = r_d. \tag{20}$$
$$M_i = 0 \Rightarrow N_i = 1; \tag{21}$$
$$H_{i,j} = 0 \Rightarrow F_{i,j} = 1; \tag{22}$$
$$H_{i,j} = 0 \Leftrightarrow H_{i',j'} = 0, E_{i',j'} = 0; \tag{23}$$
$$C_{i,j} = 1 \Leftrightarrow M_i = 1, E_{i,j} = 1, R_{i,j} = 1, F_{i,j} = 1; \tag{24}$$

Here, α_1, α_2 and α_3 are parameters of the TCM. α_1 represents the probability of $M_i = 1$ learnt from the whole task's contextual information. α_2 and α_3 are parameters which hold similar meanings.

We model Assumption 1 by (16), (21) and (24). We model Assumption 2 by (22), (23) and (24). The formula (19) models the position bias. As mentioned above, we can replace (19) with other formulas to embed existing click models into the TCM to explain the position bias. For example, if we want to integrate the UBM model, we simply need to replace (19) by

$$P(E_{i,j} = 1 | C_{i,l_j} = 1, C_{i,l_j+1:j-1} = 0) = \beta_{l_j,j} \tag{25}$$

If we want to leverage the DBN model into the TCM, we need to introduce a new class of variables, namely $S_{i,j}$, to represent the user's satisfaction. Then we can replace (19) by

$$E_{i,j-1} = 0 \Rightarrow E_{i,j} = 0$$
$$C_{i,j} = 0 \Rightarrow S_{i,j} = 0$$
$$S_{i,j-1} = 1 \Rightarrow E_{i,j} = 0$$
$$P(S_{i,j} = 1 | C_{i,j} = 1) = s_d$$
$$P(E_{i,j} = 1 | E_{i,j-1} = 0, S_{i,j-1} = 0) = \gamma$$

where s_d is the satisfaction parameter for the document retrieved in the i-th query session and ranked at the j-th position. Obviously, the TCM can be viewed as a general framework for enhancing existing click models and helping to eliminate their cross-query bias. In this paper, to focus on the demonstration of the effectiveness of the biases to interpret the task-centric user behavior, we use it in its simplest form, namely the formula from (19). The experimental results in section 6 demonstrate that even with the simple model, the TCM outperforms state-of-the-art models like DBN and UBM.

5. INFERENCE & IMPLEMENTATION

The inference algorithm estimates the value of three global parameters $\alpha_1, \alpha_2, \alpha_3$ and other probability parameters defined in the browsing model. We perform our inferences using the Expectation-Maximization (EM) algorithm. The target of EM is to maximize the log-likelihood over the log data, which is also the sum of all log-likelihoods over tasks. For each task, the TCM treats the task as a whole in which all the query sessions are mutually correlated. Thus, the likelihood of an entire task is a joint distribution over all its query sessions instead of the product of the likelihood of each individual query session. Since the basic unit of the TCM inference is the task, its implementation is more complicated than the implementation for traditional click models such as UBM and DBN.

To perform EM, we need to compute the posterior distribution for each hidden variable in the task. Such hidden variables include M_i, $E_{i,j}$ and $R_{i,j}$, which satisfy $P(M_i = 1) = \alpha_1$, $P(E_{i,j} = 1) = \beta_j$ and $P(R_{i,j} = 1) = r_d$. We also introduce two classes of auxiliary variables N'_i and $F'_{i,j}$ satisfying $P(N'_i = 1) = \alpha_2$ and $P(F'_i i, j) = \alpha_3$. Then we rewrite (17) as $N_i = 1 \Leftrightarrow M_i = 1, N'_i = 1$; and (18) as $F_{i,j} = 1 \Leftrightarrow H_{i,j} = 1, F'_{i,j} = 1$. In this way, all the parameters in the TCM can relate to their associated hidden variables. These variables are binary, mutually independent, and their probability of being 1 is equal to the value of some specific parameters.

With these hidden variables, we can perform the EM iterations. In the E-Step, we compute the marginal posterior distribution of each hidden variable to associate parameters that we introduced. The computation is performed based on the parameter values updated in the previous iteration, which are further discussed in Section 5.1. In the M-Step, all posterior probabilities associated with the same parameter are averaged to update the parameters. In particular, if we assume that H_θ is the set of hidden variables which associate with the parameter θ, we may have

$$\theta \leftarrow \frac{\sum_{X \in H_\theta} P(X = 1|D)}{|H_\theta|}. \tag{26}$$

Here, D represents the entire click-through log dataset. Actually, the posterior distribution of each hidden variable depends only on its related task.

5.1 A Fast Implementation of EM

One problem remaining is to compute the marginal posterior distributions. Unfortunately, the observed variables and hidden variables are mutually correlated in the task making the computation of marginal distributions exponentially complicated. In order to tackle the complexity problem, we have to temporally violate some specific assumptions in the TCM and assume that most variables are independent from each other. We wish to minimize the impact of this modification in the calculation of the marginal posterior. In particular, we notice that every hidden variable we consider in the TCM is closely associated with at most one document. For example, $R_{i,j}$, $E'_{i,j}$ and $F'_{i,j}$ are associated with the document located at (i, j), while M_i and N'_i have no association with any document directly. Suppose that we want to compute the marginal posterior distribution of a hidden variable X; d is the document associated with X. In this case, we preserve the relation (23) where d is located at (i, j). For all other positions where d does not appear, we redefine $H_{i,j} \equiv 0$. In this way, we consider the duplicate bias only for d and assume that there is no duplicate bias for all other documents. This modification certainly changes the TCM, but it has relatively little impact on the marginal distribution of X, because d is the only document which directly interacts with X.

Under this modification, we are able to compute more efficiently the marginal posterior distribution of X. Assume that the document d is retrieved t times in the query chain at position $(i_1, j_1), \cdots, (i_t, j_t)$. Since

$$P(X|D) = \tag{27}$$

$$\sum_{k=0}^{t} P(X, E_{i_1,j_1} = \cdots = E_{i_k,j_k} = 0, E_{i_{k+1},j_{k+1}} = 1|D)$$

we transformed the problem of computing $P(X|D)$ to the problem of computing each of the terms on the right hand of (27). Note that when the position of the first examination of d is given, all the query sessions in the task are mutually independent. Thus, every term on the right hand of (27) can be calculated in $O(tn)$ time. Because there are at most $t+1$ terms in total, the time complexity of computing $P(X|D)$ is $O(t^2 n)$.

Another problem is that the EM inference is iterative, so its efficiency is largely dependent on its convergence rate. However, the traditional EM update converges slowly in inferring click models. To see this, suppose that we want to update a relevance parameter r_d for some document d. We rewrite the update formula (26) with Bayes' Rule:

$$r_d \leftarrow \frac{\sum_{R \in H_{r_d}} P(R = 1)P(D|R = 1)/P(D)}{|H_{r_d}|}. \tag{28}$$

If we use T_R to indicate the task where R lies, then $P(D|R = 1)/P(D) = P(T_R|R = 1)/P(T_R)$. The value of $P(T_R|R = 1)$ and $P(T_R)$ can be computed following the same approximate strategy used above. Since $P(R = 1) = r$, so we have

$$r_d \leftarrow r_d \cdot \frac{\sum_{R \in H_{r_d}} P(D|R = 1)/P(D)}{|H_{r_d}|}. \tag{29}$$

The update formula (29) usually takes many iterations before r_d converges to a stable value, especially when document d is ranked in a low position which causes $\frac{P(D|R=1)}{P(D)}$ to be very close to 1. To accelerate the EM inference, we modify the update formula (28) but keep the fixed point of r_d unchanged. We note that for any relevance variable R, it must be related to an examination of some document d. There are two possibilities in this examination: either the document is clicked or it is skipped. We use $H^+_{r_d}$ to indicate the set of R where d is clicked, and use $H^-_{r_d}$ to indicate the set of R where d is skipped. If d is clicked, then we have $P(R = 1|D) = 1$; otherwise, we have

$$P(R = 1|D) = r \cdot \frac{P(D|R = 1)}{P(D)}.$$

When we substitute these two relations into (26) and replace the "$\leftarrow$" operator by an equation, we can resolve r_d as

$$r_d \leftarrow \frac{|H^+_{r_d}|}{|H^+_{r_d}| + \sum_{R \in H^-_{r_d}} P(D|R = 1)/P(D)}. \tag{30}$$

When we update r_d by formula (30), it converges much faster. In particular, if $H^+_{r_d} = \emptyset$, then r_d converges to zero immediately after the first iteration. Note that (30) is not a standard EM update, so it does not guarantee every convergence property of EM. According to our real-data experiments, however, the new update converges well in all cases.

6. EXPERIMENTS

In this section, we report the results of our experiments to demonstrate the advantages of the TCM by comparing them to the results achieved using the UBM model and the DBN model. The effectiveness of a click model is measured using both the NDCG metric [13] and the click perplexity metric.

6.1 Experimental Setup

The click logs used to develop and test the click models were collected from a popular commercial search engine for three consecutive days in December 2010. The click logs

Table 3: The summary of the data set in experiments. *Task Length* means the number of query sessions in the task. The same query may exist in tasks of different lengths.

Task Length	# Task	# Query	# Session
1	4,699,387	562,648	4,699,387
2	2,115,676	570,360	4,231,352
3	1,067,730	623,356	3,203,190
4	629,624	564,109	2,518,496
5	391,127	474,891	1,955,635
6	265,471	401,145	1,592,826
7	182,418	332,216	1,276,926
8	134,124	280,980	1,072,992
≥ 9	97,046	228,596	873,414
All	9,582,603	3,761,986	21,424,218

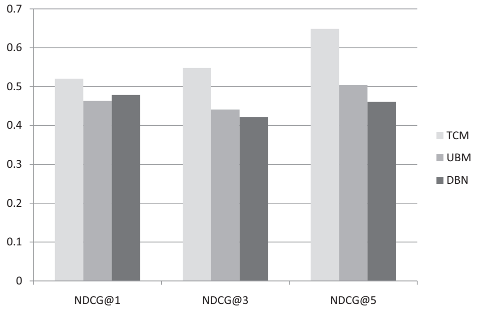

Figure 5: Overall NDCG evaluation

comprise 9.6 million search tasks and 21.4 million query sessions. The dataset is then divided randomly and evenly into a training set with 4.8 million tasks and a test set with the other 4.8 million tasks. Note that we grouped the tasks by their users in advance so that all the tasks from a user is either in the training set or the testing set. The distribution of the tasks, queries and sessions is reported in Table 3.

We also collected human editorial judgements to determine the relevance between queries and documents. This dataset is used to evaluate the effectiveness of a commercial search engine with a similar description in [21]. There are totally 17,551 queries that appear in both the click logs and the editorial judgement data set. In our experiments, we use them to calculate the *normalized discounted cumulative gain* (NDCG).

6.2 NDCG Evaluation

To evaluate relevance estimation accuracy, we trained TCM, UBM and DBN models on the same training set. The training of TCM follows the algorithm in Section 5.1. The training of UBM and DBN follows the inference algorithm introduced in the original papers. The γ parameter of DBN is set to its optimal value, namely $\gamma = 0.9$. The inference of TCM, UBM and DBN are all based on EM performed in an iterative manner. The initial value of EM parameters are set at 0.5. The iterative training has been performed until all parameters converge.

After the completed training, we obtain a relevance value for each query-document pair in the training set. We sorted the documents with respect to this estimated relevance produced by the click model and compared the ranking result with the ideal ranking derived from the editorial judgement. We then calculated the NDCG to evaluate the effectiveness of each click model.

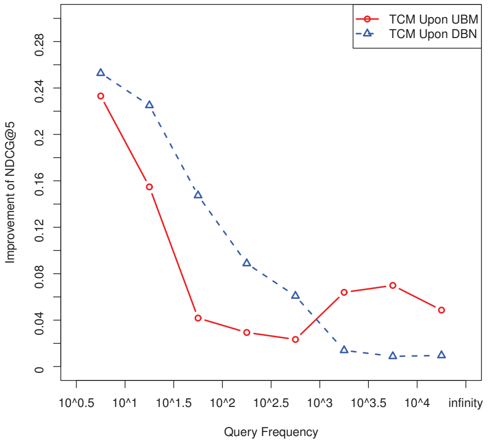

Figure 6: The NDCG@5 Improvements over query frequencies.

Table 4: The NDCG@5 scores over query types.

	Navigational Queries	Informational Queries
TCM	0.623	0.615
UBM	0.552	0.458
DBN	0.521	0.422

The NDCG scores (at different positions) for TCM, UBM and DBN are reported in Figure 5. It shows that the task-centric model significantly outperforms the traditional models. The improvements in TCM over UBM in terms of NDCG@1, NDCG@3, NDCG@5 are 0.056, 0.107, 0.145 respectively. Similarly, the improvements over DBN are 0.042, 0.127 and 0.187. It is clear that the relative improvements are all more than 10%. We also performed a t-test for the improvements and found that all the P-values of a t-test are less than 0.01%. These results verify the necessity of considering task-related biases in TCM for estimating more accurate relevance. In addition, it is worth noting that the improvements in NDCG@1 are not as significant as those in the other two positions. This might be attributed to the fact that there is no position bias in position 1. Existing models like DBN and UBM can have an acceptable performance in this position, but TCM can still show small gains over them in efficiency.

To draw a conclusion about the consistency of the improvements, we first grouped the queries with respect to their frequencies, which are the number of times a query exists in the training set. We then calculated the NDCG@5 improvements for each frequency. The results are reported in Figure 6. It is informative to note that the improvements in the low frequency queries are much more significant than those of in high frequency queries. This might demonstrate that existing click models are not effective for modeling low frequency queries due to the inefficient information. However, considering all of the task information, the TCM can determine a good relevance for these queries despite their infrequency. In Figure 6, the TCM upon UBM curve has a dip for middle frequencies. This might be attributed to the fact that UBM has a particularly good performance on middle frequency queries: it maintains the most complex transition matrix among the three models. Note that the middle frequency queries are more diversified than high frequency queries (high frequency queries like “google” and “facebook” are typically navigational) and has more dataper-query than tail queries. Thus, the structure of UBM enables it to capture more sophisticated position bias information in a single session for middle frequencies.

Second, we grouped the queries based on their *query entropy*, which characterizes the diversity of clicks on returned documents. A low entropy means that users tend to click the same document, which usually suggests a navigational query. In contrast, a high entropy indicates that the user clicks on a variety of documents usually suggests that it is an informational query. In our data set, 33.6% of the queries have entropy ranges in $[0, 0.05]$, and 28.1% of the queries have entropy ranges in $[1.6, 1.65]$. Other queries have their entropies distributed within $[0.05, 1.6]$. Therefore, we simply assume that queries in the first range are navigational queries and those in the second range are informational queries. We present the results of NDCG@5 in Table 4. It is notable that improvements in informational queries are more significant than those of navigational queries. This may be due to the well known observation that compared to navigational queries, informational queries are more ambiguous and difficult for a search engine to accurately gauge. Thus, to satisfy a search task, users have to frequently reformulate their queries to get better search results. This might bring more task information to enrich the current query. This enrichment in turn enables the TCM to learn a better relevance. To sum up, both query grouping approaches consistently verify the importance of the search task information for inferring a more accurate document relevance.

6.3 Perplexity Evaluation

In addition to NDCG, click perplexity is a widely used metric for evaluating click model accuracy. Perplexity can be seen as the log-likelihood powers which are computed independently at each position. For example, we assume that q_j^s is the probability of some click calculated from a click model, i.e. $P(C_j^s = 1)$ where C_j^s is a binary value indicating the click event at position j in query session s. Then the click perplexity at position j is computed as follows:

$$p_j = 2^{-\frac{1}{|S|}\sum_{s\in S}(C_j^s log_2 q_j^s + (1-C_j^s) log_2 (1-q_j^s))}$$

The perplexity of a data set is defined as the average of perplexities in all positions. Thus, a smaller perplexity value indicates a better consistency between the click model and the actual click data. The improvement of perplexity value p_1 over p_2 is given by $\frac{p_2 - p_1}{p_2 - 1} \times 100\%$.

As mentioned above, the two biases introduced in the TCM are supposed to better interpret user click data and learn a better relevance in training. Thus, our first experiment on perplexity is designed to verify the relevance accuracy in the test set. To remove the effect of biases in the test set to focus on pure relevance, we considered the tasks with only one query session in the test set and calculated their perplexity in prediction. In table 5, we first read the overall perplexity in the last column. It shows that the TCM can achieve 2.6% and 6.5% improvements over UBM and DBN respectively. We performed the t-test and it shows that the P-values are both less than 0.01% due to the large-scale of the dataset. We further investigated the perplexity in different ranking positions. It first showed that UBM performs better than DBN on perplexity. This is consistent with the results reported in [21]. Second, it demonstrates that the TCM achieves improvements over almost all positions. One exception happens in the lower positions (position 4 and position 5) when the TCM performance is comparable with UBM. The perplexity on this dataset is the same for these two positions. Different from the DBN model, the UBM model maintains a global matrix, such as equation (5). Thus in prediction, documents in lower positions can benefit from the click/skip information in the higher positions. The lower a position is, the more information it can obtain from higher positions. Therefore, for a document in position 4 or 5, it may have enough information to calculate the click probability: i.e., enriching more task information is no longer necessary in these positions for the UBM model, even though it is very beneficial in the high positions.

Table 5: The perplexity comparison over ranking positions. "@n" represents the perplexity at position n. "Impr." represents the improvements of TCM over UBM and DBN.

	@1	@2	@3	@4	@5	Overall
TCM	1.731	1.262	1.159	1.090	1.070	1.262
UBM	1.758	1.264	1.160	1.090	1.070	1.269
Impr.	3.5%	0.6%	0.4%	0.0%	0.0%	2.6%
DBN	1.765	1.263	1.171	1.112	1.092	1.280
Impr.	4.4%	0.1%	6.7%	19.6%	23.1%	6.5%

Table 6: The perplexity comparison over query session positions

	First Session	Second Session	Third Session	Fourth Session	Fifth Session
TCM	1.190	1.224	1.242	1.247	1.244
UBM	1.192	1.249	1.278	1.289	1.290
Impr.	1.1%	9.9%	13.0%	14.8%	15.9%
DBN	1.195	1.233	1.252	1.261	1.264
Impr.	2.6%	3.6%	4.2%	5.5%	7.6%

Next, we are interested in the relationship between the quantity of task information and the improvements in perplexity. We grouped all the query sessions based on their positions in each task and used the entire test set to calculate the perplexity. For the calculation of probability of a click in the TCM of each current query (from position 2), we used its previous queries as task information in order to help determine the "freshness" of all the documents in the current query session according to (22) and (23). We report the results in Table 6. It is clear that as the query session position rises, the improvements monotonically increase. This trend is consistent in both UBM and DBN. This leads to an intuitive conclusion that the more task information that is for available enrichment, the more significant are improvements the TCM can achieve. This observation verifies the importance of task information for inferring better relevance in click models.

7. CONCLUSION

In this paper, we have investigated the necessity of incorporating enriched task information to develop a better click model. We first used real data to prove this truth and then proposed two task-related biases to better model user click behavior under a task. We have proposed a task-centric click model (TCM) to address these two biases and illustrated its advantages in interpreting user click data. Furthermore, we have performed experiments from multiple perspectives to verify the effectiveness of the TCM. The large-scale experiments demonstrate that the TCM can consistently and significantly outperform the other two state-of-the-art models.

We believe that taking into account a wider arrange of task information is the key to learning a better click model, since a single query is often ambiguous and only imperfectly represent user intent. As demonstrated in this research, the task information is especially beneficial for low frequent (tailed) queries, which are believed to be hard to do on commercial search engines. On the other hand, the modeling approach used in this paper can possibly be applied to other search related applications with consideration of task information, such as query classifications or suggestions. Moreover, how to investigate more user personalized biases and simultaneously consider task-related biases is a promising direction for designing new click models. These are research topics for our future investigations.

Acknowledgement

We would like to thank Prof. Roger Olesen from Tsinghua University for his professional editing of the paper and Nathan Nan Liu from HKUST for his help. We also thank Hong Kong RGC Grants 621010 and N_HKUST624/09.

8. REFERENCES

[1] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 89–96, 2005.

[2] H. Cao, D. H. Hu, D. Shen, D. Jiang, J.-T. Sun, E. Chen, and Q. Yang. Context-aware query classification. In *Proceedings of the 32nd Annual ACM SIGIR Conference*, pages 3–10, 2009.

[3] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li. Context-aware query suggestion by mining click-through and session data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 875–883, 2008.

[4] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th International World Wide Web Conference*, pages 1–10, 2009.

[5] W. Chen, Z. Ji, S. Shen, and Q. Yang. A whole page click model to better interpret search engine click data. In *Proceedings of the Twenty-Fifth Conference on Artificial Intelligence*, 2011.

[6] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 1st ACM International Conference on Web Search and Data Mining*, pages 87–94, 2008.

[7] G. Dupret and C. Liao. 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*, pages 181–190, 2010.

[8] G. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. In *Proceedings of the 31st Annual ACM SIGIR Conference*, pages 331–338, 2008.

[9] T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In *Proceedings of the 27th International Conference on Machine Learning*, pages 13–20, 2010.

[10] L. A. Granka, T. Joachims, and G. Gay. Eye-tracking analysis of user behavior in www search. In *Proceedings of the 27th Annual ACM SIGIR Conference*, pages 478–479, 2004.

[11] F. Guo, C. Liu, A. Kannan, T. Minka, M. Taylor, Y. Wang, and C. Faloutsos. Click chain model in web search. In *Proceedings of the 18th International World Wide Web Conference*, pages 11–20, 2009.

[12] B. Hu, Y. Zhang, W. Chen, G. Wang, and Q. Yang. Characterize search intent diversity into click models. In *Proceedings of the 20th International World Wide Web Conference*, pages 17–26, 2011.

[13] K. Järvelin and J. Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual ACM SIGIR Conference*, pages 41–48, 2000.

[14] C. Liu, F. Guo, and C. Faloutsos. Bbm: bayesian browsing model from petabyte-scale data. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 537–546, 2009.

[15] B. Piwowarski, G. Dupret, and R. Jones. Mining user web search activity with layered bayesian networks or how to capture a click in its context. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*, pages 162–171, 2009.

[16] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th International World Wide Web Conference*, pages 521–530, 2007.

[17] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *Proceedings of the 28th Annual ACM SIGIR Conference*, pages 43–50, 2005.

[18] R. Srikant, S. Basu, N. Wang, and D. Pregibon. User browsing models: relevance versus examination. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 223–232, 2010.

[19] R. W. White, P. N. Bennett, and S. T. Dumais. Predicting short-term interests using activity-based search context. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 1009–1018, 2010.

[20] B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen, and H. Li. Context-aware ranking in web search. In *Proceedings of the 33rd Annual ACM SIGIR Conference*, pages 451–458, 2010.

[21] Y. Zhang, D. Wang, G. Wang, W. Chen, Z. Zhang, B. Hu, and L. Zhang. Learning click models via probit bayesian inference. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 439–448, 2010.

[22] Z. Zhu, W. Chen, T. Minka, C. Zhu, and Z. Chen. A novel click model and its applications to online advertising. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, pages 321–330, 2010.