A Noise-aware Click Model for Web Search

Weizhu Chen
Microsoft Research Asia
wzchen@microsoft.com

Dong Wang
Microsoft Research Asia
dongw89@gmail.com

Yuchen Zhang
Microsoft Research Asia
zhangyuc@gmail.com

Zheng Chen
Microsoft Research Asia
zhengc@microsoft.com

Adish Singla
Microsoft BING Bellevue
adishs@microsoft.com

Qiang Yang
Hong Kong University of Science & Technolgy
qyang@cse.ust.hk

ABSTRACT

Recent advances in click model have established it as an attractive approach to infer document relevance. Most of these advances consider the user click/skip behavior as binary events but neglect the context in which a click happens. We show that real click behavior in industrial search engines is often noisy and not always a good indication of relevance. For a considerable percentage of clicks, users select what turn out to be irrelevant documents and these clicks should not be directly used as evidence for relevance inference. Thus in this paper, we put forward an observation that the relevance indication degree of a click is not a constant, but can be differentiated by user preferences and the context in which the user makes her click decision. In particular, to interpret the click behavior discriminatingly, we propose a Noise-aware Click Model (NCM) by characterizing the noise degree of a click, which indicates the quality of the click for inferring relevance. Specifically, the lower the click noise is, the more important the click is in its role for relevance inference. To verify the necessity of explicitly accounting for the uninformative noise in a user click, we conducted experiments on a billion-scale dataset. Extensive experimental results demonstrate that as compared with two state-of-the-art click models in Web Search, NCM can better interpret user click behavior and achieve significant improvements in terms of both perplexity and NDCG.

Categories and Subject Descriptors

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

General Terms

Algorithms, Measurement, Experimentation

Keywords

click model, click noise, log analysis

WSDM'12, February 8–12, 2012, Seattle, Washington, USA.

1. INTRODUCTION

Modeling user behavior recorded in search engine click-through logs is attracting more and more attention in information retrieval research, since user click behavior seems to be an excellent source of encoding user preferences to search results. Furthermore, in industrial search engines, click-through logs can be collected at a very low cost. This makes the modeling scalable and practical to better understand user favorites. Consequently, many attempts have already positioned it as an appealing area and formalized this issue of learning document relevance from click-through logs as a click modeling problem.

Although click-through logs are very informative, a well-known challenge for click modeling is position bias, where it states that a document appearing in a higher position is more likely to attract more user clicks even though it is not as relevant as other documents appearing in lower positions. Thus, the often used metric click-through rate (CTR) is not an exact measure of document relevance. An effective click model needs to alleviate position bias. This bias was firstly noticed by Granka et al. [11] in their eye-tracking experiments and a lot of research since then has proposed to correct it and infer an unbiased relevance. Thereafter, Richardson et al. [18] proposed to increase the relevance of documents in lower positions by a multiplicative factor; Craswell et al. [8] later formalized this idea as the examination hypothesis which states that a document is clicked if and only if it is both examined and relevant.

Under the examination hypothesis, given a document that has been examined, its relevance is exclusively determined by its CTR. Thus, for each click in the logs, it contributes to the CTR and directly plays a positive impact on increasing the relevance of a clicked document. Perversely, after carefully examining the real user click behavior in an industrial search engine, we observed that user click behavior is often complex and noisy. It is far from ideal that every click is informative and can be a good indication of relevance. Rather, we show that there is a considerable amount of noise in clicks and some clicked documents turn out to be irrelevant documents. Obviously, the existence of click noise may hinder click models to infer an accurate document relevance. Yet, most of the existing works on click model only treat each click as a binary event (click or not) but disregard the noise or quality born with it. This may result in their research being based on unreliable user click signals for relevance inference.

In this paper, we put forward an observation that not

all clicks are equal and not all clicks are good indications of relevance. Rather, we illustrate the complex causes of a click and the necessity to characterize the noise in each click. To capture the click noise, we propose a Bayesian model called Noise-aware Click Model (NCM). NCM is designed to complement the click data with the human judged data so as to learn a predictor to characterize click noise. NCM is also capable of predicting the noise of large-scale clicks without human judged data by exploiting the user preferences and the context in which a user makes her click decision. Different from previous research in click modeling, NCM assesses the noise of each click and encourages the high quality clicks with less noise to play an important role in relevance inference. In addition, NCM is the first attempt to incorporate both click data and human labeled data together in learning a click model thus can be regarded as a semi-supervised model. We developed NCM as a general model which makes it capable of embracing the assumptions of most existing click models. Especially, we successfully extended NCM to embrace the assumptions of two state-of-the-art models in Web search. To verify the effectiveness of NCM, we conducted experiments on a billion-scale industrial dataset. Extensive experimental results demonstrated that NCM can achieve consistent and significant improvements in terms of both perplexity and NDCG.

The rest of the paper is organized as follows. In Section 2, we briefly introduce the preliminaries of click models and summarize the related work. We introduce the methodology in Section 3 and the NCM in Section 4. Finally, we conduct comparison experiments in Section 5 and conclude the paper in Section 6.

2. PRELIMINARIES & RELATED WORKS

Before delving into hypothesis details, we begin by introducing some definitions and background that will be used throughout the paper. A user starts a *query session* by issuing a *query* to a search engine, the search engine returns M (usually $M = 10$) ranked *documents* in a *Search Engine Result Page* (SERP). We use $d_{\phi(i)}$ to indicate the document ranked at the position i. We assume all documents are indexed and d_i is the i-th document. Here we use a mapping function ϕ to represent the document at position i. The user then examines the SERP and clicks on some or none of the documents. Clicks on sponsored ads or other web elements such as query suggestion results are not considered in our query session. Any subsequent query re-submission or re-formulation will be regarded as initiating a new query session. It is worth noting that there exist different definitions of a user session in prominent literature. Thus in this paper, we differentiate them by query session and search session, where a query session only contains the actions related to a single query, while a search session includes all the actions a user undertakes to perform a search task, which may include one or more query sessions, multiple query reformulations and clicks under different queries.

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

2.1 Examination Hypothesis

The *examination hypothesis* assumes that a displayed document is clicked if and only if this document is both *examined* and *relevant*. In the literature[6, 20, 25], if the document is examined, the perceived relevance of a document is a query-specific variable which directly measures the likelihood that a user will click this document. More precisely, 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 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 query q and 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.

Following the examination hypothesis, given the condition $P(E_i = 1)$, 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 click-through 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 latest clicked position l_i in the same query 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 $C_i = C_{i+1} = \cdots = C_j = 0$.

A similar work to UBM is Bayesian browsing model (BBM) [17], 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)[28]. It extends the model to consider multiple biases and shows that previous models are special cases of GCM.

2.2 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:

$$P(E_1 = 1) = 1 \quad (8)$$
$$P(E_{i+1} = 1|E_i = 0) = 0 \quad (9)$$
$$P(C_i = 1|E_i = 1) = a_{\phi(i)} \quad (10)$$
$$P(E_{i+1} = 1|E_i = 1, C_i) = 1 - C_i \quad (11)$$

Eq.11 implies that a user will abandon the query session if she finds a desired search result; otherwise she always continues the examination. This simultaneously reveals that it can only be applied to query sessions with one click at most. Yet this is too strict for real logs with multiple clicks in a query session.

Subsequently, the DCM, CCM and DBN are introduced to deal with the multiple clicks within a query session. The *dependent click model* (DCM) [13] introduces a set of global position-dependent parameters to represent the probability of examining the next document after a click. *Click Chain Model*(CCM) [12] continues to model the relationship between the examination probability and the relevance of previous documents. The *Dynamic Bayesian Network* (DBN) [6] model emphasizes that a click does not necessarily indicate user's satisfaction with the document. Instead, the user may have been attracted by some misleading snippets, including the title and summary, to trigger a click. Hence, it distinguishes document relevance as *perceived relevance* a_i and *actual relevance* s_i. Whether a user clicks a document or not depends on its perceived relevance and whether the user is satisfied with the document or not depends on the actual relevance. If the user is satisfied with the clicked document, she will not examine the next document. Otherwise, there is a probability $1 - \gamma$ that the user abandons her query session and a probability γ that the user continues her search. Thus, DBN replaces Eq.11 as follows:

$$P(S_i = 1|C_i = 0) = 0 \quad (12)$$
$$P(S_i = 1|C_i = 1) = s_{\phi(i)} \quad (13)$$
$$P(E_{i+1} = 1|S_i = 1) = 0 \quad (14)$$
$$P(E_{i+1} = 1|E_i = 1, S_i = 0) = \gamma \quad (15)$$

Where S_i is a hidden event indicating whether a user is satisfied with the document $d_{\phi(i)}$. The values a_i and s_i are estimated by applying the expectation-maximization algorithm in the original paper, while there exists a probit approach to infer the model introduced in [26].

2.3 Externalities and Others

Unlike the UBM model and the other models following the cascade assumption, which assumes that the examination event will only be affected by the documents shown above, externalities consider that the click behavior in a position will be simultaneously affected by the documents below. Two recent works [20][24] have conducted good experiments to demonstrate this phenomenon in online advertising and developed models to characterize it. [24] verified the existing of externalities based on two advertisements and modeled their competitive property as attracting a user for a click. Work[20] studied the externalities in online advertisements with respect to the clicks in other below positions and stated that the relevance of a document is not a constant but affected by clicks in other positions. This factor was also observed in an experimental finding in a previous work [4]. It is pointed out that the CTR of an advertisement can be affected by the quality of other advertisements shown together. It is obvious that externalities exist in online advertising and most of the previous works are motivated by observations from the advertisement instead of general Web Search. As compared with the general Web Search, the number of advertisements in a SERP is smaller and users may have a more commercial intent when clicking on an advertisement. This may encourage users to conduct more comparison between advertisements.

There are four other models that are not part of previous assumptions. The whole page click model (WPC)[7] interprets user click behavior in the whole page, including both the search and ads. The post-click click model (PCC)[27] captures user post-click behavior after the click. The Session Utility Model (SUM) [10], given a single query, measures the relevance of a set of clicked documents as the probability that a user stops the query session. The intent-bias model [14] demonstrates the existence of multiple intents for a same query and extends UBM and DBN to characterize the diversity of search intents in click models.

2.4 Modeling Context

Contextual information has been utilized by many applications and previous works have already demonstrated its usefulness in understanding user behavior. [1] used the contextual information as features to improve Web Search ranking. [22] represented the contextual information as ODP categories and used them to predict user short-term interests. [23] considered how users reformulate queries and used this information for Web search ranking. [3] leveraged the contextual information to better model query completion. [9] utilized the contextual user behavior in organic search to characterize the user behavior in sponsored search. [19] used the contextual information for recommendation systems and showed its effectiveness in collaborating filtering problem. Our work differs from these studies. First, we are working on a click model problem to automatically infer document relevance based on user click behavior. Second, we focus on utilizing the contextual behavior and user preference to understand click noise and then qualify a click for inferring a more accurate relevance.

3. THE DATA & METHODOLOGY

Previous works have already illustrated the complexity of user behavior and the challenges to interpret click data [22, 19], since the cause for a user to perform a click may be complex and varied. She may be attracted by a snippet, or just want to explore some information need with a strong uncertainty. Whenever we are aiming to interpret the complex user click data in a commercial search engine, there is no doubts that click data are not extremely clean. To let the real data speak for this, we first collect a human judged dataset containing 474,185 judged documents and collect 16.18 million clicks on these documents. We then classify all these clicks based on the relevance rating of judged documents, from *Bad, Fair, Good, Excellent* to *Perfect*. We present the relationship between the number of clicks and each corresponding rating in Figure 1. The x-axis indicates the relevance degree from irrelevant to the most relevant and the y-axis illustrates the density distribution of this click data. It clearly shows that there are less than 35% clicks on perfect documents. If we treat *Bad* and *Fair* in judged ratings as irrelevant, there are more than 28% of

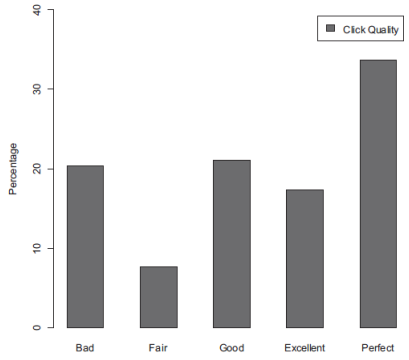

Figure 1: The distribution of click quality.

clicks on irrelevant documents. This clearly shows that there exists a considerable amount of noise in click data, i.e, not all clicks are good indications of document relevance.

To characterize the noise of a click, we complement click data with human judged data. The human judged dataset is collected from a commercial human relevance system (HRS), where it randomly picks up a set of representative queries and requires judges to give a rating denoting the relevance between a query and each of its corresponding documents. If we treat this human judged data as ground-truth, we may classify all the clicks on judged documents into two categories. For a click on a document which is judged as relevant in the HRS system, we define it as a noise-free click; otherwise it could be a noisy click since the click might not bring a good indication of relevance. If we can have the judged rating for each clicked document, we can identify the noise of each click easily. However, the size of the queries in the HRS system is very limited due to the high cost of manual judgement. Even if we are using a HRS set which is used to evaluate the performance of a commercial search engine, it only contains 12,590 queries. While in the click through data for one month, there are billions of queries and clicks. Apparently, compared with the HRS data, click data are much larger in scale and can be collected with a much lower cost. Given the HRS data as the ground-truth, this can be vividly represented as a semi-supervised learning problem where small portion of click data are labeled but most data are unlabeled. Hence, a big challenge is to generalize the limited HRS labeled data to understand the noise in each click, and then we may use it to infer a more accurate relevance via click models for large-scale unlabeled queries.

Our methodology is to design a model to leverage the user preference and contextual information as features for this generalization. We list them in Table 1, which can be classified into two categories. The *Context* class specifies the contextual information in which users make the click decision, such as its previous queries and click information in these queries. The *User* class characterizes the historical behavior of a user, such as her average click or skip behavior for documents in other search sessions. Based on the limited labeled documents in HRS, we classify all the clicks on them into two categories: noise-free click and noisy click. The task in the training is to learn a predictor to characterize the relationship between values of the features mentioned above and the degree of a noise. With this predictor learnt in the training phrase, we may predict the noisy degree for large-scale unlabeled click data. The challenge lies in how to learn this predictor effectively with consideration of the assumptions in click models. The details will be presented in next section.

An interesting analysis is to understand the cause of the noise although it may be complex and varied. A well known explanation is the difference between perceived relevance and intrinsic relevance[10] , where perceived relevance corresponds to a snippet while intrinsic relevance relates to a document. While a user performs a click, she is mostly attracted by a snippet but is generally not aware of whether the corresponding document is relevant or not. Similarly, as demonstrated in [21], there is a strong disagreement between the judgement of snippet and that of the document itself. A snippet may make an irrelevant document appear relevant, or a relevant document appear irrelevant. However, a click happens before a user examines the document thus it may have some disagreement with the intrinsic relevance. This disagreement will be characterized in our Noise-aware click model.

Based on the data we use, one other cause of the noise may be the difference between general search users' conception of relevance and the judges', especially when search users have multiple intents for the same query, such as an informational query with an ambiguous meaning. However, it is very hard to ask a judge to understand each intent for each query. We are aware of this possible cause and that numerous studies have documented this difference [5]. Thus in a commercial HRS system, it adopts a multiple-judges approach to alleviate this for ambiguous queries. For each ambiguous query, it asks about three judges to give ratings for all its corresponding documents based on as many intents as they can be aware of. This approach may most likely reduce the inconsistency and cover most intents from general users. Another issue caused by the multiple judges is their mutual disagreement. We studied a dataset and found that if we calculate from the 5-level ratings:*Bad, Fair, Good, Excellent* and *Perfect* respectively, the pair-wise consistency is about 70%. However, if we only separate the ratings into binary categories with *Bad, Fair* as irrelevant while the others as relevant, the pair-wise consistency value increases to more than 90%. This mean that, in most of the data, judges can agree with each other; for the remaining small portion of data with inconsistent ratings, we may use the majority of this binary rating as the final rating.

Another cause of this noise may be related to the type of queries. Given a set of navigational queries, users may have a fairly consistent and clear idea of what they are looking for, so the clicks among different users may be consistent and informative for relevance inference. While for a navigational query, users might represent more exploratory information needs or situations where there is an uncertainty and lack of domain knowledge. This may lead to more inconsistent click behavior among users and more clicks on irrelevant documents. This query type difference will be modeled by our noise-aware click model via a query specific parameter.

4. NOISE-AWARE CLICK MODEL

In this section, we first present the model specifications of NCM and then illustrate its extension to embrace the assumptions of two state-of-the-art click models in Web Search. Following this we introduce the inference of NCM to make

Feature Name	Feature description
User class	
AvgDwellTime	The average dwell time of clicks for this user.
IntervalTime	The average interval time between two clicks for this user.
UserSkip	True if the user skips this document before.
UserClick	True if the user clicks this document before.
UserFirstClick	True if the user first clicks this document before.
UserLastClick	True if the user last clicks this document before.
UserOnlyClick	True if the user only clicks this document before.
FracQueryNoClick	Fraction of queries with no clicks for this user.
FracQueryOneClick	Fraction of queries with one click for this user.
FracQueryMultiClicks	Fraction of queries with more than one clicks for this user.
Context class	
SubmitTime	The issued time of current query.
QuerySubsetPre	True if current query is a subset of previous query.
QuerySupersetPre	True if current query is a superset of previous query.
QueryDistPre	Edit distance between current query and previous query.
ClickInLastSession	Whether there is a click in previous query session.
DwellTimeInLastSession	The dwell time the user spends on last session (in seconds).
FirstQuery	True if it is the first query in the search session.
TimeInSearch	Time spent on the search engine so far (in seconds).
URLInSearch	Number of URL in the search session so far.
QueryInSession	Number of queries in the search session so far.
ClickInSession	Number of clicks in the search session so far.
AvgTimeBetQueries	Average interval time between two issued queries.
TimeToLastAction	Time to last action such as submit a query or click.

Table 1: Features used in NCM.

it capable of learning from large-scale data and present its prediction formula.

4.1 Model

The Noise-aware click model (NCM) introduces a random variable N to characterize the noise degree of a click. There are two extreme contexts for a click. In an extremely noisy context with $N = 1$, users may click on irrelevant documents. While in a noise-free context, users tend to click on relevant documents. With the limited HRS data as ground-truth to represent a relevant or irrelevant rating of a document, we may make the following two observations on this noise factor:

- In a noise-free context with $N = 0$, users tend to skip (i.e., not click) a document with a *Bad* or *Fair* rating.
- In an extremely noisy context with $N = 1$, a click tends to be a noisy event that is not fully dependent on document relevance.

Figure 2 is a NCM flow chart that illustrates the user behavior when examining a document at position i. It characterizes two extreme contexts: The extremely noisy context with $N_i = 1$ and the noise-free context with $N_i = 0$. Suppose document $d_{\phi(i)}$ is relevant to query q according to the HRS rating. In a noise-free context, a user will examine it and then click it with probability $r_{\phi(i)}$. Here we do not assume $r_{\phi(i)} = 1$ because this value is to be learnt via click models. Also, if the document is irrelevant according to the HRS rating, a user in a noise-free context tends to skip it. On the other hand, in an extremely noisy context, the user may be attracted by its snippet but click it according to a parameter b no matter whether the document is relevant or not, where b is a query-specific parameter. In either way,

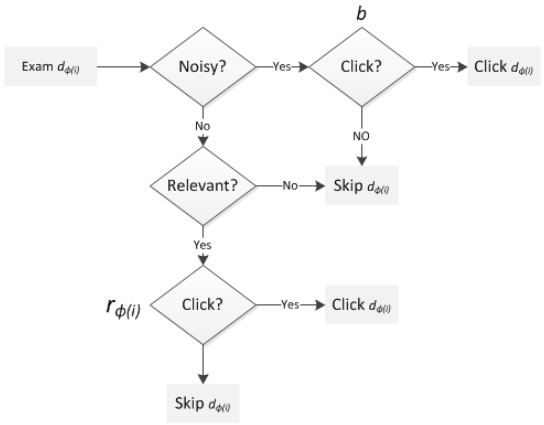

Figure 2: The user click behavior of NCM

after the user makes her decision, she can choose to continue the examination of other documents or abandon the current query session according to the assumptions defined in traditional click models. It is worth noting that these two extreme contexts are only formalized for the limited clicks with HRS ratings. Meanwhile, we are to learn the noise degree predictor by using the labeled documents in HRS with features extracted from their corresponding clicks as the training data. Thereafter, for the unlabeled clicks without HRS ratings, the degree of noise will be predicted using the predictor, i.e, calculating $P(N_i = 1)$ according to the contextual feature values.

To represent our assumptions in a probabilistic way, we use a symbol L_i to represent the binary rating of the $i-$th document. $L_i = 1$ indicates that the document at position

i has a relevant HRS rating to the query otherwise $L_i = 0$. We use N_i to represent the degree of noise in the context when a user is examining the document at position i. $N_i = 0$ indicates that the click happens in a noise-free context while $N_i = 1$ indicates that the context is extremely noisy. Under these definitions, we use the following equations to represent the above flow chart.

$$P(N_i = 1) = g(f_1, f_2, \cdots, f_n) \quad (16)$$
$$P(C_i = 1|E_i = 0) = 0 \quad (17)$$
$$P(C_i = 1|E_i = 1, L_i = 1, N_i = 0) = r_{\phi(i)} \quad (18)$$
$$P(C_i = 1|E_i = 1, L_i = 0, N_i = 0) = 0 \quad (19)$$
$$P(C_i = 1|E_i = 1, N_i = 1) = b \quad (20)$$

In the above, $r_{\phi(i)}$ indicates the click probability when a user in a noise-free context examining the document at position i and this document has a relevant HRS rating with $L_i = 1$. The f_i represents each user behavior feature defined in Table 1 and $g : R^n \to R$ is a function that maps features to a value indicating how noisy a context is. There are many kinds of functions such as the logistic function and sigmoid function which can be used to represent g. In our work, we define $g(f_i) = \Phi(\sum w_i f_i)$, where w_i is the feature weight. $\Phi(x) = \int_{-\infty}^{x} N(t; 0, 1)dt$ is the cumulative distribution function of the standard normal distribution. It is also referred to as the probit link [2]. We use it for convenience of inference and to assure the probability value within an interval of $[0, 1]$. We divide continuous feature f_i into several buckets so that we only need to consider binary features in the inference step. Generally, the estimated relevance in our model is defined as the probability of a click in a noise-free context given the document has been examined.

The NCM model is general and can function under many click models. Since it does not make any assumptions or constraints in estimating the probability of examination, it can be used with existing models based on the examination hypothesis. In this paper, we apply it to two typical click models, UBM and DBN, and adopt a Bayesian inference method to learn the parameters.

4.2 Noise-aware UBM

Same as the NCM specifications, the Noise-aware UBM model will introduce a random variable N_i to characterize the noise of the context where a click happens. We denote the noise-aware UBM as N-UBM with the following equations:

$$P(E_i = 1|C_{1:i-1} = 0) = \beta_{0,i} \quad (21)$$
$$P(E_i = 1|C_{l_i} = 1, C_{l_i+1:i-1} = 0) = \beta_{l_i,i} \quad (22)$$
$$P(N_i = 1) = \Phi(\sum w_i f_i) \quad (23)$$
$$P(C_i = 1|E_i = 0) = 0 \quad (24)$$
$$P(C_i = 1|E_i = 1, L_i = 1, N_i = 0) = r_{\phi(i)} \quad (25)$$
$$P(C_i = 1|E_i = 1, L_i = 0, N_i = 0) = 0 \quad (26)$$
$$P(C_i = 1|E_i = 1, N_i = 1) = b \quad (27)$$

In N-UBM, the probability of $N_i = 1$ is calculated from the probit function based on the feature values related with the context. If a user is in a noise-free context where $N_i = 0$ and the document is with a relevant rating with $L_i = 1$, the click probability is determined by the examination probability $P(E_i = 1)$ and a relevance parameter $r_{\phi(i)}$. But if the document has an irrelevant rating where $L_i = 0$ under a noisy-free context where $N_i = 0$, the click probability is assumed to be 0. On the other hand, in an extremely noisy context with $N_i = 1$, N-UBM assumes that a user will click the document according to a query specific parameter b, which is not characterized in traditional UBM assumptions. For the value of $P(N_i = 1)$, it will be calculated based on the contextual feature values f_i and the learnt weights w_i, which will be discussed in Section 4.4. After the user finishes her action in this position, she will continue to examine the following documents with probabilities related to β.

4.3 Noise-aware DBN

Similar to N-UBM model, the N-DBN model introduces a random variable N_i to characterize the noise of a context. When a user is in a noisy-free context, she will click the relevant document ($L_i = 1$) and skip the irrelevant document ($L_i = 0$) given that the document has been examined. If the user is in an extremely noisy context, there is a query-specific probability b that the document will be clicked, but this click event is not dependent on the document relevance. If the document is not clicked, there is a probability γ to examine the next document and $1 - \gamma$ to abandon the search. We can formalize the N-DBN model by following equations:

$$P(E_1 = 1) = 1 \quad (28)$$
$$P(C_i = 1|E_i = 0) = 0 \quad (29)$$
$$P(N_i = 1) = \Phi(\sum w_i f_i) \quad (30)$$
$$P(C_i = 1|E_i = 1, L_i = 1, N_i = 0) = r_{\phi(i)} \quad (31)$$
$$P(C_i = 1|E_i = 1, L_i = 0, N_i = 0) = 0 \quad (32)$$
$$P(C_i = 1|E_i = 1, N_i = 1) = b \quad (33)$$
$$P(S_i = 1|C_i = 0) = 0 \quad (34)$$
$$P(S_i = 1|C_i = 1) = s_{\phi(i)} \quad (35)$$
$$P(E_{i+1} = 1|E_i = 0) = 0 \quad (36)$$
$$P(E_{i+1} = 1|S_i = 1) = 0 \quad (37)$$
$$P(E_{i+1} = 1|E_i = 1, S_i = 0) = \gamma \quad (38)$$

4.4 Inference and Implementation

The parameter estimation of NCM is a two-stage procedure due to the existance of HRS data. In the first stage, we use the HRS data and the limited click data on the labeled documents to train a noise predictor to characterize the degree of noise for a click, i.e. we estimate the value of w_i $(i = 1, \cdots, n)$. The predictor takes a feature vector $\mathbf{f}$ and outputs a noise probability, i.e. $\Phi(\mathbf{w}^T\mathbf{f})$. Note that the inference in this stage only involves the sessions for a limited number of queries whose corresponding documents have been judged in the HRS system.

In the second stage, we infer the click model on all search sessions in the click-through logs. The training no longer depends on the HRS judgements but moves on to all unlabeled queries. We use the noise predictor obtained in the previous stage and the contextual feature values to estimate the noise degree of a context in which a click happens. This allows us to more accurately simulate real user behavior and learn the model parameters accordingly, such as the relevance parameters for each query-document pair. When the training is completed, we are able to predict the noise degree of clicks in future sessions based on both w_i and related contextual feature values f_i $(i = 1, \cdots, n)$. Next, we illustrate the im-

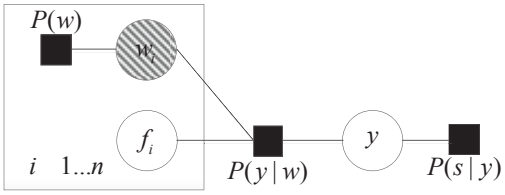

Figure 3: Factor graph for updating parameters w_i.

plementation of the two inference stages and the prediction formula.

4.4.1 Stage 1: Training with HRS

In the first stage, the training data is the intersection of click-through logs and HRS judgements. In other words, we consider only the queries and documents for which the relevance is judged. Given the noise-aware click model defined in Section 4.1, let $\mathbf{w}$ and $\mathbf{\Theta}$ be the parameters of the model: $\mathbf{w} = (w_1, \cdots, w_n)$ are coefficients of the noise predictor and $\mathbf{\Theta} = (\theta_1, \cdots, \theta_m)$ are other probability parameters, including the relevance of each query-document pair and the parameters with respect to model assumptions. We employ an approximate Bayesian inference to estimate $\mathbf{w}$ and $\mathbf{\Theta}$. The learning process is incremental: We load and process search sessions one by one, and the data for each session is discarded after they have been processed. Given a new incoming session s, we update the distribution of each parameter based on the session data and the click model. Before the update, $\mathbf{w}$ and $\mathbf{\Theta}$ have prior distributions $p(\mathbf{w})$ and $p(\mathbf{\Theta})$. We compute the likelihood function $P(s|w_i)$ and $P(s|\theta_j)$, multiply each to the prior distribution and then derive the posterior distribution $p(w_i|s)$ and $p(\theta_j|s)$. Finally, these posteriors are used as priors in the processing of next session.

With the Probit Bayesian Inference (PBI) technique invented by Zhang et. al [26], we can smoothly update the distribution of each θ_j. The update of each w_i is somewhat complicated because w_i is a real number instead of a probability, which is incompatible with the PBI's input format. However, the problem can be solved by introducing an auxiliary variable $y = \mathbf{w}^T\mathbf{f}$. Since $\Phi(y)$ is a probability, we first apply PBI to derive the posterior distribution $p(\Phi(y)|s)$, and then calculate $p(y|s)$. Actually, PBI guarantees that $p(y|s)$ is always a Gaussian density. Furthermore, each $p(w_i|s)$ is the marginal of the joint distribution $p(y, \mathbf{w}|s)$ after integrating y and all other variables in $\mathbf{w}$, denoted as $\mathbf{w}^{\backslash i}$, so we can calculate it by the following integration:

$$p(w_i|s) \propto \int (\prod_{i=1}^{n} p(w_i)) p(y|\mathbf{w}) p(s|y) \mathrm{d}\mathbf{w}^{\backslash i} \mathrm{d}y; \quad (39)$$

$$p(s|y) \propto \frac{p(y|s)}{p(y)} = \frac{p(y|s)}{\int (\prod_{i=1}^{n} p(w_i)) p(y|\mathbf{w}) \mathrm{d}\mathbf{w}} \quad (40)$$

The Eq. (39) and Eq.(40) can be efficiently computed by the sum-product message passing algorithm on the factor graph (Figure 3), where (39) represents the leftward message passing and (40) represents the rightward message passing. A more detailed illustration of the message passing algorithm can be found in [16].

When the training is completed, we can get a distribution for each w_i. The inference procedure ensures that all distributions are Gaussian, so we can assume that $p(w_i) = \mathcal{N}(w_i; \mu_i, \sigma_i^2)$. Thus, given an arbitrary feature vector $\mathbf{f}$, we can predict the probability of noise by the expectation of $\Phi(\mathbf{w}^T\mathbf{f})$:

$$P(N = 1) = E(\Phi(\mathbf{w}^T\mathbf{f})) = \Phi\Big(\frac{\sum_{i=1}^{n} \mu_i f_i}{\sqrt{1 + \sum_{i=1}^{n} \sigma_i^2 f_i^2}}\Big).$$

4.4.2 Stage 2: Training without HRS

In the second stage, the training data are the entire click-through logs that mostly consist of unlabeled queries and their corresponding documents. For the unlabeled click data with L_i unknown in this stage, we formulate the noise-aware click probability as:

$$P(C_i = 1|E_i = 0) = 0; \quad (41)$$

$$P(C_i = 1|E_i = 1) = P(N_i = 0) r_{\phi(i)} + P(N_i = 1) b; \quad (42)$$

Here, $P(N_i = 0) = 1 - P(N_i = 1)$ is given by the predictor trained in Stage 1. Since parameter $r_{\phi(i)}$ is exactly the click-through rate of $d_{\phi(i)}$ after examined by a user in a noise-free context, it is the value of the document relevance and will be used for predicting future clicks in Section 4.5

Again, we employ the Probit Bayesian Inference (PBI) to perform parameter estimation. The inference is simpler than that in Stage 1 since w_i are constant values at this stage, and all other parameters are probability values. Thus, we use PBI to go through the whole data set and derive the probability distribution for each parameter. In most cases, the variance of such distribution converges to zero, so we get a numerical estimation after the training. If the variance is still large, we compute the expectation of the parameter to achieve a numerical estimation. The detailed formulas are also given in [26].

4.5 Prediction

Given a test session, the probability distribution of a click event in this session at position i can be calculated by the formula below:

$$P(C_i = 1) = P(E_i = 1)((1 - \Phi(\sum w_i f_i)) r_{\phi(i)} + \Phi(\sum w_i f_i) b)$$

In N-DBN, $P(E_i = 1)$ is calculated as the below equations:

$$P(E_1 = 1) = 1 \quad (43)$$

$$P(E_{i+1} = 1) = (1 - P(C_i = 1) s_{\phi(i)}) \gamma \quad (44)$$

In calculating $P(E_i = 1)$ for N-UBM, we need to consider the probability of clicks in each position above it. If there exists at least one click above, it enumerates the latest click position j above the position i. Then, we can calculate it by the equation below:

$$P(E_i = 1) = P(C_{1:(i-1)} = 0)\beta_{0,i} + \sum_{j=1}^{i-1} P(C_j = 1) P(C_{(j+1):(i-1)} = 0) \beta_{j,i} \quad (45)$$

5. EXPERIMENTAL RESULTS

In this section, we evaluate the NCM model by comparing it with two state-of-the-art click models in Web search. As we implement our NCM to embrace the assumptions of these two models, the reformulated models, i.e., the Noise-aware DBN and the Noise-aware UBM, are denoted as N-DBN and N-UBM respectively. We use *perplexity* and *normalized discounted cumulative gains (NDCG)* [15] to evaluate the performance of different click models.

5.1 Experimental Data

Click log data: The query sessions used to train and evaluate the click models are collected from a commercial search engine in the U.S. market in English from January 1st to January 31st in 2011. Several query sessions from one user may belong to the same search session, thus we uses a 30 minutes inactivity time interval to separate any two search sessions. For each query session, we extract contextual information from other query sessions appearing before and in the same search session. We collect the values for each feature in Table 1. In order to prevent the whole dataset being dominated by some extremely frequent queries, we limit the number of query sessions for each query to 10^4. When calculating perplexity, we use the first 15 days of data as the training data and the rest 16 days of data as the test data. In total, we collect approximately 380 million distinct queries and 1.17 billion query sessions. The detailed information about the dataset is summarized in Table 2.

HRS data: The details of the HRS data have already been introduced in Section 3. We provide its distribution over query frequency in table 2. Overall, there are 12,150 queries which appear in both the HRS data and the click data. For each query, there are 39 judged documents on average. While we are training the NCM in the stage 1, we treat the rating of *Bad* or *Fair* as irrelevant, and the other three ratings (*Good, Excellent* and *Perfect*) as relevant.

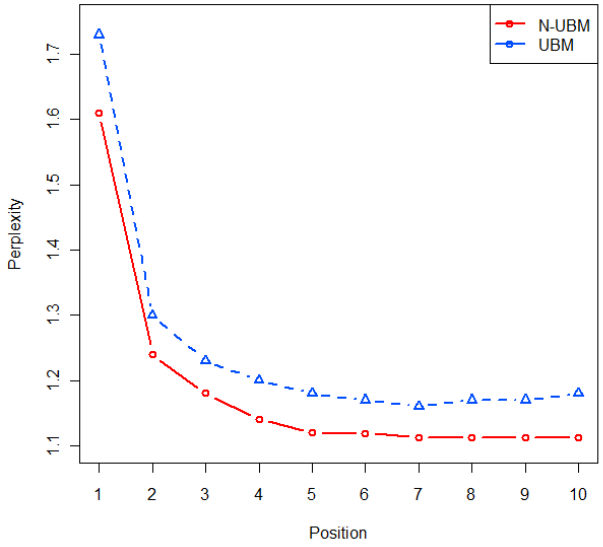

Figure 4: N-UBM & UBM perplexity Over Position

5.2 Perplexity

After the click model estimates its parameters while in training, as illustrated in Section 4.5, we can predict the click probabilities in the test query sessions. We evaluate the prediction accuracy by click perplexity, which has been widely used to measure the quality of click models[8, 12, 28]. A smaller perplexity indicates a better prediction accuracy and the optimal value is 1.0. For a given position i and a set of query 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 query 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:

$$p_i = 2^{-\frac{1}{n}\sum_{i=1}^{n}(c_i \log_2 q_i + (1-c_i)\log_2 q_i)} \tag{46}$$

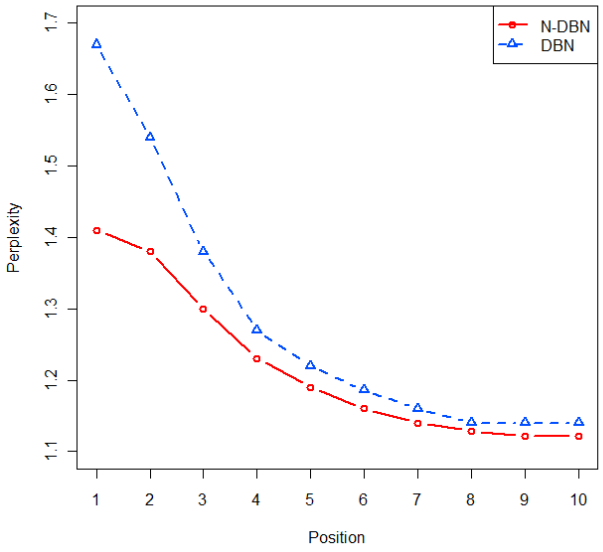

Figure 5: N-DBN & DBN perplexity Over Position

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 $(p_b - p_a)/(p_b - 1) \times 100\%$ [12].

In Figure 4 and 5, we report the perplexity of N-DBN vs. DBN, and N-UBM vs. UBM on the testing dataset over different positions. The experimental results clearly show that N-UBM and N-DBN consistently outperform the original UBM and DBN respectively. We also provide the overall results in Table 3. The overall relative improvements of N-DBN over DBN and N-UBM over UBM are 30.4% and 33.9% respectively. This is particularly notable given that the dataset we used is a billion-scale one from a commercial search engine. Next, we separate queries into six groups as Table 2 based on the number of times that each query appears in the training sessions, calculate the overall perplexity for each group, and show the results in Figure 6. If we look at the two solid lines, it clearly shows that the N-DBN consistently outperforms DBN at all frequency level. Meanwhile, if we look at the two dash lines, N-UBM also consistently performs better than UBM. It is worth noting that the relative improvement of the high frequency group is much larger than that of the low frequency group due to a difference in dominator in the calculation. For example, the relative improvement of N-DBN over DBN in the first group with the lowest frequency is 20.4% while that of the last group with the highest frequency is 48.9%.

5.3 Ranking Performance

In this part of experiments, we sort the documents with respect to the estimated relevance given by a click model, and compare the ranking result with the ideal ranking in the HRS judgement data. In this case, the relative order of estimated relevance is more important than value.

We use NDCG to evaluate our ranking results. NDCG is a well-known metric which measures the gap between the currently available ranking and the theoretically ideal ranking.

Query Frequency	#HRS Query	#HRS Rating	#Query in Train	#Session in Train	#Query in Test	#Session in Test
$1 \sim 3$	1,726	33,079	157,202,923	179,610,940	181,477,994	207,287,519
$3 \sim 10^1$	2,329	57,201	18,777,342	84,904,526	21,617,259	97,766,874
$10^1 \sim 10^2$	3,858	106,082	3,688,839	93,883,180	4,275,037	109,035,850
$10^2 \sim 10^3$	2,439	118,224	332,590	84,461,513	386,995	97,924,566
$10^3 \sim 10^4$	1,089	86,970	30,794	76,442,296	35,363	87,544,334
$> 10^4$	719	72,629	2729	27,290,000	2985	29,850,000
Total	12,150	474,185	180,032,217	546,592,455	207,795,633	629,409,143

Table 2: The summary of the data set collected from one month in Jan 2011.

Model		Perplexity	NDCG@1	NDCG@2	NDCG@3	NDCG@4	NDCG@5
DBN	N-DBN	1.2182	0.738	0.724	0.727	0.740	0.753
	DBN	1.2846	0.651	0.641	0.648	0.656	0.663
	Improvement	30.4%	13.2%	12.9%	12.3%	12.6%	13.5%
UBM	N-UBM	1.1859	0.700	0.698	0.707	0.714	0.718
	UBM	1.2490	0.609	0.618	0.631	0.643	0.650
	Improvement	33.9%	14.9%	13.0%	12.0%	11.1%	10.5%

Table 3: Detailed experimental results for different click models.

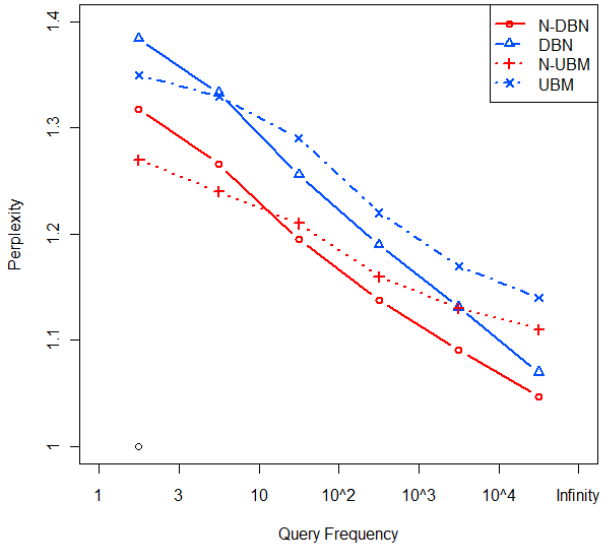

Figure 6: Perplexity Over Frequency

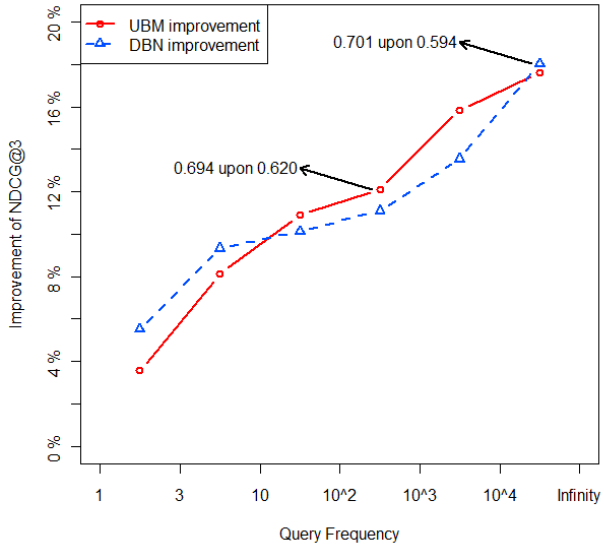

Figure 7: Imp of NDCG@3 over query frequency.

For a given query q, the results are sorted by the decreasing order of estimated relevance, and then the NDCG is calculated as

$$NDCG@i = \frac{1}{IdealDCG@i} \sum_{j=1}^{i} \frac{2^{g_i} - 1}{\log 1 + j}, \qquad (47)$$

where $IdealDCG@i$ is the maximum DCG over all possible rankings. g_i is the HRS judgement of the i-th document. It represents judges' rating on the relevance of the document to q. Note that for those documents which have no rating, we treat them as $g_i = 0$. For the whole HRS data, we randomly pick half of the queries into the training of the stage 1 and leave the other half of the queries as the evaluation queries to calculate the NDCG.

We show the detailed NDCG results (at different positions) for DBN, N-DBN, UBM and N-UBM in Table 3. The results show that the noise-aware models have significant improvements over the original models in terms of NDCG. The improvements at all positions are over 10%. These results demonstrate that after taking click noise into consideration by the NCM model, clicks models can infer a more accurate document relevance. We perform a t-test to verify the significance. The result shows that the P-values of t-test are all less than 0.01, which confirms that the estimated relevance from N-DBN and N-UBM is consistently and significantly better than that from DBN and UBM respectively.

To investigate the variance of NDCG over different query frequencies, we recalculate the NDCG@3 improvements over frequency. The results are shown in Figure 7. This experiment continues to verify the consistent improvements over different frequency. The improvement of high frequency group is more significant than that of the low frequency group. This may be attributed to the property that high-frequency queries can usually provide more sufficient information for the NCM model to better understand click noise.

6. CONCLUSION

In this paper, we have introduced a noise-aware model to capture the noisiness of a click which causes the learning model to weight differently the user click observation. We design the NCM by complementing click data with the HRS data and characterizing the context in which a user performs her click decision. NCM is a general model to embrace the assumptions in most existing click models. We have successfully extended it to embrace the assumptions of two typical click models and formalized the new models as N-UBM and N-DBN. We have designed a Bayesian inference approach to make NCM capable of processing large-scale click data. The billion-scale experimental results demonstrate the necessity to account for uninformative click noise and identify good click data in learning a click model.

NCM is initially designed as a semi-supervised model to learn the click data with limited human labeled data. It is actually a very general model to learn for two inconsistent objectives, like the HRS based objective and click likelihood based objective in this paper. This two-stage approach in NCM is also suitable for addressing the problem of a limited number of samples for one objective, like the HRS based objective with a limited size.

7. ACKNOWLEDGMENTS

We thank the support of Hong Kong RGC GRF Grant 621010. We thank Si Shen and Nan Liu from HKUST and the anonymous reviewers for their help and comments.

8. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. SIGIR '06, pages 19–26.

[2] J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

[3] Z. Bar-Yossef and N. Kraus. Context-sensitive query auto-completion. WWW '11, pages 107–116. ACM.

[4] H. Becker, C. Meek, and D. Chickering. Modeling contextual factors of click rates. AAAI '07, pages 1310–1315.

[5] P. Borlund. The concept of relevance in IR. *Journal of the American Society for information Science and Technology*, 54(10):913–925, 2003.

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

[7] W. Chen, Z. Ji, S. Shen, and Q. Yang. A whole page click model to better interpret search engine click data. In *AAAI*, 2011.

[8] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. WSDM '08, pages 87–94.

[9] C. Danescu-Niculescu-Mizil, A. Broder, E. Gabrilovich, V. Josifovski, and B. Pang. Competing for users' attention: on the interplay between organic and sponsored search results. WWW '10, pages 291–300.

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

[11] L. A. Granka, T. Joachims, and G. Gay. Eye-tracking analysis of user behavior in www search. SIGIR '04, pages 478–479.

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

[13] F. Guo, C. Liu, and Y. M. Wang. Efficient multiple-click models in web search. WSDM '09, pages 124–131.

[14] B. Hu, Y. Zhang, W. Chen, G. Wang, and Q. Yang. Characterize Search Intent Diversity into Click Models. WWW '11, pages 17–26.

[15] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20:422–446, October.

[16] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Throry*, (498-519), 1998.

[17] C. Liu, F. Guo, and C. Faloutsos. BBM: Bayesian browsing model from petabyte-scale data. SIGKDD '09, pages 537–546. ACM.

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

[19] H. Z. S.H. Yang, B. Long, A. Smola and Z. Zheng. Collaborative Competitive Filtering : Learning Recommender Using Context of User Choice. SIGIR '11.

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

[21] A. Turpin, F. Scholer, K. Jarvelin, M. Wu, and J. Culpepper. Including summaries in system evaluation. SIGIR '09, pages 508–515. ACM.

[22] R. W. White, P. N. Bennett, and S. T. Dumais. Predicting short-term interests using activity-based search context. CIKM '10, pages 1009–1018.

[23] B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen, and H. Li. Context-aware ranking in web search. SIGIR '10, pages 451–458.

[24] W. Xu, E. Manavoglu, and E. Cantu-Paz. Temporal click model for sponsored search. In *SIGIR*, pages 106–113. ACM, 2010.

[25] Y. Zhang, W. Chen, D. Wang, and Q. Yang. User-click modeling for understanding and predicting search-behavior. In *KDD*, 2011.

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

[27] F. Zhong, D. Wang, G. Wang, W. Chen, Y. Zhang, Z. Chen, and H. Wang. Incorporating post-click behaviors into a click model. In *SIGIR*, 2010.

[28] Z. A. Zhu, W. Chen, T. Minka, C. Zhu, and Z. Chen. A novel click model and its applications to online advertising. WSDM '10, pages 321–330.