Incorporating Post-Click Behaviors into a Click Model

Feimin Zhong[12], Dong Wang[12], Gang Wang[1], Weizhu Chen[1],Yuchen Zhang[12],
Zheng Chen[1],Haixun Wang[1]
[1] Microsoft Research Asia, Beijing, China
[2]Tsinghua University, Beijing, China
{v-fezhon, v-dongmw, gawa, wzchen, v-yuczha, zhengc, haixunw } @microsoft.com

ABSTRACT

Much work has attempted to model a user's click-through behavior by mining the click logs. The task is not trivial due to the well-known position bias problem. Some breakthroughs have been made: two newly proposed click models, DBN and CCM, addressed this problem and improved document relevance estimation. However, to further improve the estimation, we need a model that can capture more sophisticated user behaviors. In particular, after clicking a search result, a user's behavior (such as the dwell time on the clicked document, and whether there are further clicks on the clicked document) can be highly indicative of the relevance of the document. Unfortunately, such measures have not been incorporated in previous click models. In this paper, we introduce a novel click model, called the post-click click model (PCC), which provides an unbiased estimation of document relevance through leveraging both click behaviors on the search page and post-click behaviors beyond the search page. The PCC model is based on the Bayesian approach, and because of its incremental nature, it is highly scalable to large scale and constantly growing log data. Extensive experimental results illustrate that the proposed method significantly outperforms the state of the art methods merely relying on click logs.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation, Performance

Keywords

Post-Click Behavior, click log analysis, Bayesian model

1. INTRODUCTION

It is one of the most important as well as challenging tasks to develop an ideal ranking function for commercial search engine. Most of existing works depend on manually labeled data, where professional editors provide the relevance ratings between a query and its related documents. According to manually labeled data, machine learning algorithms [5, 10, 13] are used to automatically optimize the ranking function and maximize user satisfaction. However, the labeled data is very expensive to be generated and is difficult to keep up with the trend over time. For example, given a query "SIGIR", a search engine is expected to return the most up-to-date site such as the SIGIR 2010 website to users, instead of SIGIR 2009. Thus, it is very difficult to maintain the relevance labels up to date.

Compared with manually labeled data, terabytes of implicit user clicks are recorded by commercial search engines every day, which implies that a large scale of click-through data can be collected at a very low cost and it usually reveals the latest tendency of the Internet users. User preference on search results is encoded into user clicks, as such, the click logs provide a highly complementary information to manually labeled data. Many studies have attempted to discover the underlying user preferences from the click-through logs and then learn a ranking function, or regard the click logs as a complementary data source to overcome shortcomings in manually labeled data. Following the pioneered works by Joachims et al.[14] that automatically generated the preferences from the click logs to train a ranking function, many interesting works have been proposed to estimate the document relevance from user clicks, including [1, 2, 3, 6, 18].

Previous works have noticed that the main difficulty in estimating the relevance from click data comes from the so-called position bias: a document appearing in a higher position is more likely to attract user clicks even though it is irrelevant. Recently, Richardson et al.[19] suggested to reward the document relevance at a lower position by multiplying a factor and this idea was later formalized as the examination hypothsis [8] and the position model [7], which indicates the user will click a document only after examining it. Craswell et al. [8] extended the examination hypothesis and proposed the cascade model by assuming that the user will scan search results from top to bottom. Furthermore, Dupret and Piwowarski[9] included the positional distance into the proposed UBM model. Guo et al.[11] proposed the CCM model and Chappell and Zhang[7] proposed the DBN model that generalizes the cascade model by introducing that the conditional probability of examining the current document is related to the relevance of the document at the previous position.

Despite their successes in solving the position-bias prob-

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.
Copyright 2010 ACM 978-1-60558-896-4/10/07 ...$10.00.

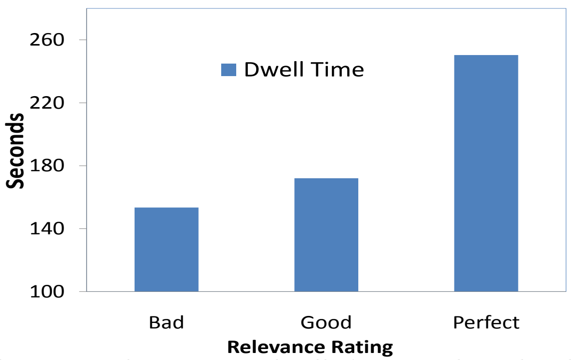

Figure 1: The average dwell time on three levels of relevance rating.

lem, previous works mainly investigate user behaviors on the search page, without considering user subsequent behaviors after a click. Nevertheless, as pointed in the DBN model, a click only represents user is attracted by the search snippet, rather than indicates the clicked document is relevant or user is satisfied with the document. Although there is a correlation between clicks and document relevance, they often differ with each other in many cases. For example, given two documents with similar clicks, if users often dwell longer to read the first document while close the second document immediately, it is likely that users feel satisfied with the first document while disappointed with the second one. Obviously, the relevance difference between these two documents can be discovered from user post-click behaviors, such as the dwell time on the clicked document. As shown in Figure 1, we calculate the average dwell time on three relevance levels in a manually labeled data set[1] It is clear that there is a strong correlation between the dwell time and the relevance rating, which validates the importance of incorporating user post-click behaviors to build a better click model.

User subsequent behaviors after a click have been studied for evaluating and improving the quality of the results returned by search engine. Sculley et al.[20] attempted to predict the bounce rates and Attenberg et al.[4] attempted to predict expected on-site actions in sponsored search. Agichtein et al.[2] optimized the ranking function through including some features extracted from post-click behaviors. Post-click behaviors can act as an effective measure of user satisfaction, thus, are very useful to improve the ranking function. However, there are few works investigate how to integrate both click behaviors and post-click behaviors into a click model.

In this paper, we propose a novel click model, called post-clicked click model (PCC), to provide an unbiased estimation of the relevance from both clicks and post-click behaviors. In order to overcome the position bias in clicks, the PCC model follows the assumptions in the DBN model [7] that distinguishes the concepts of the perceived relevance and the actual relevance. It assumes that the probability that user clicks on a document after examination is determined by the perceived relevance, while the probability that user examines the next document after a click is determined by the actual relevance of the previous document. Different from DBN, the post-click behaviors are used to estimate the user satisfaction in the PCC model. Some measures such as the user dwell time on the clicked page, whether user has the next click, etc are extracted from the post-click behaviors, and used as features that are shared across queries in the PCC model.

The PCC model is based on the Bayesian framework that is both scalable and incremental to handle the computational challenges in the large scale and constantly growing log data. The parameters for the posterior distribution can be updated in a closed form equation. We conduct extensive experimental studies on the data set with 54931 distinct queries and 140 million click sessions. Manually labeled data is used as the ground truth to evaluate the PCC model. The experimental results demonstrate that the PCC model significantly outperform two state of the art methods such as the DBN and CCM models that do not take post-click behaviors into account. Because the PCC model can provide much more number of accurate preference data complementary to manually labeled data, the ranking function trained on the relevance labels from both the PCC model and manually labeled data can produce better NDCG value than merely trained on manually labeled data.

2. PRELIMINARIES

We firstly introduce some background before delving into the algorithm details. When a user submits a query to the search engine, the search engine returns the user some ranked documents as search results. The user then browses the returned documents and clicks some of them. One query session corresponds to all the behaviors the user does under one input query, and we assume there are M displayed documents in each query session.

2.1 Examination and Cascade Hypotheses

The studies on click model attempted to solve the click bias problem in user implicit feedback. There are two important hypotheses, i.e., the examination hypothesis and the cascade hypothesis, that are widely used in various click model implementations. These two hypotheses are quite natural to simulate user browsing habits, and our proposed PCC model also depends on them.

We use two binary random variables E_i and C_i to represent the examination and click events of the document at the position i $(i = 1, ..., M)$. $E_i = 1$ indicates the document at the position i is examined by the user, while $E_i = 0$ indicates this document is not examined. $C_i = 1$ indicates the user clicks the document at the position i, while $C_i = 0$ indicates the user does not click this document.

The examination hypothesis assumes that when a displayed document is clicked if and only if this document is both examined and perceived relevant, which can be summarized as follows:

$$\begin{aligned} P(C_i = 1 \mid E_i = 0) &= 0 && (1) \\ P(C_i = 1 \mid E_i = 1) &= a_{u_i}, && (2) \end{aligned}$$

where u_i is the document at the position i, and the parameter a_{u_i} measures the relevance[2] of the document u_i indicating the conditional probability of click after examination.

The cascade hypothesis assumes that the user scans linear to the search results, thus, a document is examined only if all the above documents are examined. The first document

[1] The data set information is introduced in Section 4.

[2] a_{u_i} is the perceived relevance in the DBN model

is always examined.

$$P(E_{i+1} = 1 \mid E_i = 0) = 0 \quad (3)$$
$$P(E_1 = 1) = 1. \quad (4)$$

2.2 DBN Click Model

Since the proposed model follows similar assumptions in the DBN model, we briefly introduce the formulation in DBN. A click does not necessarily indicates that the user is satisfied with this document. Thus, the DBN model [7] distinguish the document relevance as the perceived relevance and the real relevance, where whether the user clicks a document depends on its perceived relevance while whether the user is satisfied with this document and examines the next document depends on the real relevance. Thus, besides the examination and the cascade hypotheses, the DBN click model is characterized as:

$$P(S_i = 1 | C_i = 1) = s_{u_i} \quad (5)$$
$$C_i = 0 \Rightarrow S_i = 0 \quad (6)$$
$$S_i = 1 \Rightarrow E_{i+1} = 0 \quad (7)$$
$$P(E_{i+1} = 1 | E_i = 1, S_i = 0) = \gamma, \quad (8)$$

where S_i is a binary variable indicating whether the user is satisfied with the document u_i at the position i, and the parameter s_{u_i} measures the real relevance of this document. The DBN model uses the EM algorithm to find the maximum likelihood estimation of the parameters.

2.3 Post-Click Behaviors

Behavior logs in this study are the anonymized logs provided by users who opted in through a widely-distributed browse toolbar. These log entities include a unique anonymous identifier for the user, the issued query to search engine, the visited document, and a timestamp for each page view or search query.

We process behavior logs, and extract all the post-click behaviors after there is a document click on the search page, Thus, for each pair of query and document, several behavior sessions from different users are extracted and the length of each session is fixed no longer than 20 minutes. We then define some measures extracted from the post-click sessions:

- *Dwell time on the next clicked page*;
- *Dwell time on the clicked pages in the same domain*;
- *Interval time that user inputs another query*;
- *Whether user has the next click on the clicked document*;
- *Whether user switches to another search engine.*

For each query and document pair, we calcuate the average value of the above measures over related sessions and the averaged values are used as features into the proposed algorithm.

3. POST-CLICKED CLICK MODEL

We now introduce a novel model, post-clicked click model (PCC), that leverages both click-through behaviors on the search page and the post-click behaviors after the click.

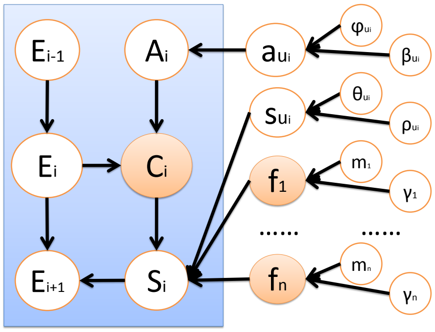

Figure 2: The PCC model. The variables C_i and f_i ($\forall i$) are the observed variables given a query session.

3.1 Model

The PCC model is a generative Bayesian model and is explained in Figure 2, where the variables inside the box are defined at the session level, and the variables outside are defined at the query level. The variables E_i, C_i, and S_i are defined the same as in the Section 2. Here we assume there are n features extracted from user post-click behaviors and f_i is the feature value of the ith feature.

$$a_u \sim N(\varphi_u, \beta_u^2), s_u \sim N(\theta_u, \rho_u^2), f_i \sim N(m_i, \gamma_i^2). \quad (9)$$

Thus, φ_u and β_u^2 are the parameters of the perceive relevance variable a_u, θ_u and ρ_u^2 are the parameters of the real relevance variable s_u, and m_i and γ_i^2 are the parameters of the ith feature variable f_i.

The PCC model is characterized by the following equations:

$$E_1 = 1 \quad (10)$$
$$A_i = 1, E_i = 1 \Leftrightarrow C_i = 1 \quad (11)$$
$$P(A_i = 1 \mid E_i = 1) = P(a_u + \epsilon > 0) \quad (12)$$
$$P(S_i = 1 \mid C_i = 1) = P(s_u + \textstyle\sum_{i=1}^{n} y_{u,i} f_i + \epsilon > 0) \quad (13)$$
$$C_i = 0 \Rightarrow S_i = 0 \quad (14)$$
$$S_i = 1 \Rightarrow E_{i+1} = 0 \quad (15)$$
$$P(E_{i+1} = 1 \mid E_i = 1, S_i = 0) = \lambda \quad (16)$$
$$E_i = 0 \Rightarrow E_{i+1} = 0, \quad (17)$$

where $\epsilon \sim N(0, \beta^2)$ is an error parameter and $y_{u,i}$ is a binary value indicating whether we can extract the value of the ith feature on the document u. It is possible that, for a document u, no user has clicked this document, thus, there is no information extracted from post-click behaviors on the ith feature. Thus, $y_{u,i} = 0$ in this case. Otherwise, $y_{u,i} = 1$.

The PCC model simulates user interactions with the search engine results. When a user examines the ith document, he will read the title and the snippet of this document, and whether the document attracts him depends on the perceived relevance of this document a_{u_i}. If the user is not attracted by the snippet (i.e., $A_i = 0$), he will not click the document which also indicates he is not satisfied with this document (i.e., $S_i = 0$). Thus, there is a probability λ that the user will examine the next document at the position $i+1$, and a probability $1 - \lambda$ that the user stops his search on this query. If the user is attracted by the snippet (i.e., $A_i = 1$), he will click and visit the document. User post-click behaviors on the clicked document are very indicative to infer how much the user is satisfied with this document. If the user is satisfied (i.e., $S_i = 1$), he will stop this search session; Otherwise, he will either stop this search session or examine the next document depending on the probability λ.

The equations (10) and (17) is the cascade hypothesis and the equation (11) is the examination hypothesis. The equation (12) shows that when a user examines the document, whether the user would click or not depends on the variable a_{u_i} and the error term. The equation (13) shows that when the user clicks and visits the document, the value of the post-click behavior features will affect whether the user is satisfied or not. The equation (14) and (15) mean that the user will not be satisfied if he does not click the document, while the user will stop the search when he is satisfied. The equation (16) shows that if user is not satisfied by the clicked document, the probability he continues browsing the next search results is λ while the probability he abandons the session is $1 - \lambda$.

3.2 The Parameter Update

After observing one query session, we update the related parameters of each document in this session. For each document in one query session, it can be distinguished into five cases and the parameter update for these five cases are different. We denote l as the last clicked position. When $l = 0$, it corresponds to the session with no click, and when $l > 0$, it corresponds to the session with clicks. We define two sets of positions: $\mathcal{A}$ is the set of positions before the last click and $\mathcal{B}$ is the set of positions after the last click. Thus, the five cases are defined as follows:

- Case 1 : $l = 0$, which indicates there is no click in the session. In this case, we update the parameters of the kth document with the equation (23).
- Case 2 : $l > 0, k \in \mathcal{A}, C_k = 0$, which indicates the kth document is at the non-clicked position before the last click. In this case, we update the parameters with the equation (24).
- Case 3 : $l > 0, k \in \mathcal{A}, C_k = 1$, which indicates the kth document is at the clicked postion before the last click. In this case, we update the parameters with the equations (25) , (26) and (27).
- Case 4 : $l > 0, k = l, C_k = 1$, which indicates the kth document is at the last clicked position. In this case, we update the parameters with the equations (28) , (29) and (30).
- Case 5 : $l > 0; k \in \mathcal{B}, C_k = 0$, which indicates the kth document is at the position after the last click. In this case, we update the parameters with the equation (31).

For a fixed $k (1 \leq k \leq M)$, suppose x is the parameter we want to update, we follow the equation:

$$p(x \mid C^{1:k}) \propto p(x) \times P(C^{1:k} \mid x) \tag{18}$$

to get the posterior distribution. Then we approximate it to Gaussian distribution use KL-divergence. The method to derive the updating formula is based on the message passing [15] and the expectation propagation[17]. Since the space limitation, we omit the proof of these formula. For convenience, we will introduce some functions that will be used in the following update equations:

$$N(c) = \frac{1}{2\pi} e^{\frac{-c^2}{2}}; \tag{19}$$

$$\Phi(c) = \int_{-\infty}^{c} N(x) dx; \tag{20}$$

$$v(c, \omega) = \frac{N(c)}{\Phi(c) + \frac{\omega}{1-\omega}}; \tag{21}$$

$$w(c, \omega) = v(c, \omega)(v(c, \omega) + c). \tag{22}$$

3.2.1 *Case 1:*

For the kth document, the observation is $A_1 = 0, E_1 = 1, C_i = 0, \quad 1 \leq i \leq k$. We update the parameters related to the ith document. This is the update of the parameter in the perceived relevance:

$$\begin{cases} \varphi_{u_k} \leftarrow \varphi_{u_k} - \frac{\beta_{u_k}^2 v(c, \omega_{1,k})}{(\beta^2 + \beta_{u_k}^2)^{\frac{1}{2}}} \\ \beta_{u_k}^2 \leftarrow \beta_{u_k}^2 (1 - \frac{\beta_{u_k}^2 w(c, \omega_{1,k})}{\beta^2 + \beta_{u_k}^2}) \\ c = -\frac{\varphi_{u_k}}{(\beta^2 + \beta_{u_k}^2)^{\frac{1}{2}}} \end{cases} \tag{23}$$

where $\omega_{1,k}$ is a coefficient whose value is given in Appendix. The parameters of the features and the real relevance are kept the same.

3.2.2 *Case 2:*

For the kth document, the observation is $A_k = 0, E_k = 1$. Thus, we update the parameters related to the kth document. The update of the parameter in the perceived relevance is:

$$\begin{cases} \varphi_{u_k} \leftarrow \varphi_{u_k} - \frac{v(c,0) \beta_{u_k}^2}{(\beta_{u_k}^2 + \beta^2)^{\frac{1}{2}}} \\ \beta_{u_k}^2 \leftarrow \beta_{u_k}^2 (1 - \frac{\beta_{u_k}^2 w(c,0)}{\beta_{u_k}^2 + \beta^2}) \\ c = \frac{-\varphi_{u_k}}{(\beta_{u_k}^2 + \beta^2)^{\frac{1}{2}}}. \end{cases} \tag{24}$$

The parameters of the features and the real relevance are kept the same.

3.2.3 *Case 3:*

For the kth document, the observation is $A_k = 1, E_k = 1$ and $S_k = 0$. Thus, we update the parameters related to the kth document. The update of the parameter in the perceived relevance is:

$$\begin{cases} \varphi_{u_k} \leftarrow \varphi_{u_k} + \frac{v(c,0) \beta_{u_k}^2}{(\beta_{u_k}^2 + \beta^2)^{\frac{1}{2}}} \\ \beta_{u_k}^2 \leftarrow \beta_{u_k}^2 (1 - \frac{\beta_{u_k}^2 w(c,0)}{\beta_{u_k}^2 + \beta^2}) \\ c = \frac{\varphi_{u_k}}{(\beta_{u_k}^2 + \beta^2)^{\frac{1}{2}}}. \end{cases} \tag{25}$$

The update of the parameter in the feature is:

$$\begin{cases} m_i \leftarrow m_i - \frac{v(c,0) \gamma_i^2 y_{u_k,i}}{(\sum_{j=1}^n y_{u_k,j} \gamma_j^2 + \rho_{u_k}^2 + \beta^2)^{\frac{1}{2}}} \\ \gamma_i^2 \leftarrow \gamma_i^2 (1 - \frac{\gamma_i^2 w(c,0) y_{u_k,i}}{\sum_{j=1}^n y_{u_k,j} \gamma_j^2 + \rho_{u_k}^2 + \beta^2}). \\ c = \frac{-(\theta_{u_k} + \sum_{j=1}^n y_{u_k,j} m_j)}{(\sum_{j=1}^n y_{u_k,j} \gamma_j^2 + \rho_{u_k}^2 + \beta^2)^{\frac{1}{2}}} \end{cases} \tag{26}$$

The update of the parameter in the real relevance is:

$$\begin{cases} \theta_{u_k} \leftarrow \theta_{u_k} - \frac{v(c,0)\rho_{u_k}^2}{(\sum_{j=1}^{n} y_{u_k,j}\gamma_j^2+\rho_{u_k}^2+\beta^2)^{\frac{1}{2}}} \\ \rho_{u_k}^2 \leftarrow \rho_{u_k}^2(1-\frac{\rho_{u_k}^2 w(c,0)}{\sum_{j=1}^{n} y_{u_k,j}\gamma_j^2+\rho_{u_k}^2+\beta^2}) \\ c = \frac{-(\theta_{u_k}+\sum_{j=1}^{n} y_{u_k,j}m_j)}{(\sum_{j=1}^{n} y_{u_k,j}\gamma_j^2+\rho_{u_k}^2+\beta^2)^{\frac{1}{2}}} \end{cases} \tag{27}$$

3.2.4 *Case 4*

For the last clicked document, the observation is $C_l = 1, C_i = 0(i = l+1 \quad to \quad M)$ and we update the parameters related to the lth document. The update of the parameters in the perceived relevance is:

$$\begin{cases} \varphi_{u_l} \leftarrow \varphi_{u_l} + \frac{v(c,0)\beta_{u_l}^2}{(\beta_{u_l}^2+\beta^2)^{\frac{1}{2}}} \\ \beta_{u_l}^2 \leftarrow \beta_{u_l}^2(1-\frac{\beta_{u_l}^2 w(c,0)}{\beta_{u_l}^2+\beta^2}) \\ c = \frac{\varphi_{u_l}}{(\beta_{u_l}^2+\beta^2)^{\frac{1}{2}}}. \end{cases} \tag{28}$$

The update of the parameters in the feature is:

$$\begin{cases} m_i \leftarrow m_i + \frac{v(c,\omega_2)\gamma_i^2}{(\sum_{j=1}^{n} y_{u_l,j}\gamma_j^2+\rho_{u_l}^2+\beta^2)^{\frac{1}{2}}} \\ \gamma_i^2 \leftarrow \gamma_i^2(1-\frac{\gamma_i^2 w(c,\omega_2)}{\sum_{j=1}^{n} y_{u_l,j}\gamma_j^2+\rho_{u_l}^2+\beta^2}) \\ c = \frac{(\theta_{u_l}+\sum_{j=1}^{n} y_{u_l,j}m_j)}{(\sum_{j=1}^{n} y_{u_l,j}\gamma_j^2+\rho_{u_l}^2+\beta^2)^{\frac{1}{2}}} \end{cases} \tag{29}$$

where ω_2 is a coefficient whose value is given in Appendix.

The update of the parameters in the real relevance is:

$$\begin{cases} \theta_{u_l} \leftarrow \theta_{u_l} + \frac{v(c,\omega_2)\rho_{u_l}^2}{(\sum_{j=1}^{n} y_{u_l,j}\gamma_j^2+\rho_{u_l}^2+\beta^2)^{\frac{1}{2}}} \\ \rho_{u_l}^2 \leftarrow \rho_{u_l}^2(1-\frac{\rho_{u_l}^2 w(c,\omega_2)}{\sum_{j=1}^{n} y_{u_l,j}\gamma_j^2+\rho_{u_l}^2+\beta^2}) \\ c = \frac{(\theta_{u_l}+\sum_{j=1}^{n} y_{u_l,j}m_j)}{(\sum_{j=1}^{n} y_{u_l,j}\gamma_j^2+\rho_{u_l}^2+\beta^2)^{\frac{1}{2}}} \end{cases} \tag{30}$$

3.2.5 *Case 5*

For the kth document, the observation is $C_l = 1, C_k = 0(k = l+1 \quad to \quad M)$. Thus we update the parameter related to the kth document. The update of the parameter in the perceived relevance is:

$$\begin{cases} \varphi_{u_i} \leftarrow \varphi_{u_i} - \frac{\beta_{u_i}^2 v(c,\omega_{3,k})}{(\beta^2+\beta_{u_i}^2)^{\frac{1}{2}}} \\ \beta_{u_i}^2 \leftarrow \beta_{u_i}^2(1-\frac{\beta_{u_i}^2 w(c,\omega_{3,k})}{\beta^2+\beta_{u_i}^2}) \\ c = -\frac{\varphi_{u_i}}{(\beta^2+\beta_{u_i}^2)^{\frac{1}{2}}} \end{cases} \tag{31}$$

where $\omega_{3,k}$ is a coefficient whose value is given in Appendix. The parameters in the features and the real relevance are kept the same.

3.3 Algorithm

Following the above update formula, we can easily build the PCC training algorithm as follows:

```
Initialize $a_u$, $f_i$ and $s_u$ ($\forall u, i$) to the prior distribution $N(-0.5, 0.5)$.
For each session
    If $l = 0$, update each document with (23)
    Else
        For $k = 1 \quad to \quad M$
            If $k < l$ , $C_k = 0$, update (24)
            If $k < l$ , $C_k = 1$, update (25),(26) and (27)
            If $k = l$, update (28),(29) and (30)
            If $k > l$, update (31)
        Endfor
    Endif
End
```

Given a collection of training sessions, we sequentially update the parameters according to the five cases. Since the update formula is in a closed form, the algorithm can be trained on a large scale and constantly growing log data. After training the PCC model, we set the user satisfaction probability to zero, i.e., $P(S = 1 \mid C = 1) = 0$, for those documents that have never been clicked.

The PCC model follows the assumption in DBN to distinguish the document relevance as the perceived relevance $P(A = 1|E = 1)$ and the real relevance $P(S = 1|C = 1)$. We define the document relevance inferred from the PCC model as:

$$\begin{aligned} rel_u &= P(A=1 \mid E=1)P(S_u=1 \mid C=1) \\ &= \Phi(\frac{\varphi_u}{(\beta_u^2+\beta^2)^{\frac{1}{2}}})\Phi(\frac{\theta_u+\sum_{i=1}^{n} y_{u,i}m_i}{(\rho_u^2+\beta^2+\sum_{i=1}^{n} y_{u,i}\gamma_i^2)^{\frac{1}{2}}}). \end{aligned} \tag{32}$$

This document relevance rel_u will be evaluated on the ground truth ratings in manually labeled data.

4. EXPERIMENTAL RESULTS

In the experiment, we evaluate the document relevance and the click perplexity inferred from the PCC model, and the results are compared with other click models including DBN and CCM. The experiments are organized into four parts. In the first part, we analyze the pairwise accuracies of the relevance among different click models. In the second part, we use the generated relevance to rank the documents directly and evaluate the ranking function according to the normalized discounted cumulative gain (NDCG) [12]. In the second part, we use the RankNet algorithm to learn a ranking function on the preference pairs extracted from both the click model and manually labeled data, and illustrate the ranking improvement. Finally, we illustrate the click perplexity among different click models.

4.1 Data Set

The click logs used to train the click models are collected from a large commercial search engine which comprises 54,931 randomly sampled queries and about 2 million related documents from the U.S. market in English language, and the total number of search sessions from one month click-through log is about 143 million. For each search session, we have one input query, a list of returned documents on browsed pages and a list of positions of the clicked documents. The information on the click logs is summarized in Table 1.

For each query and document pair, we collect corresponding post-click sessions in 20 minutes from one month behavior log. We calculate the average values of five features, as introduced in Section 2.3, from post-click behaviors and they are used to train and evaluate the PCC model.

The manully labeled data is used as the ground truth to evaluate the relevance from click models. In the human relevance system (HRS), editors provided the relevance ratings for 4,521 queries and 127,519 related documents. On average, 28.2 documents per query are labeled. A five grade

Query Frequency	♯ Query	♯ Document	♯ Total Sessions
1 to 30	33,519	437,610	182,312
30 to 100	5,836	163,133	332,194
100 to 1,000	8,270	425,594	3,031,827
1,000 to 10,000	5,282	578,198	17,827,303
>10,000	2,024	401,083	121,589,355
all	54,931	2,005,618	142,962,991

Table 1: The summary of the search sessions from one month click logs.

rating is assigned to each query and document (4: perfect, 3: excellence, 2: good, 1: fair, 0: bad). The documents without judgement are labeled as 0. The summary of the HRS is introduced in Table 2.

Query Frequency	♯ Query	♯ Document
1 to 30	772	11,328
30 to 100	666	12,335
100 to 1,000	1,342	33,568
1,000 to 10,000	1,074	37,092
>10,000	662	33,196
all	4,516	127,519

Table 2: The summary of the data in human relevance system (HRS).

4.2 Pairwise Accuracy

The document relevance is derived from the PCC model according to Equation (32), and we compute the relevance for those queries and related documents that are overlapped with the HRS data in the experiment. Since the relevance value is a real number between $[0, 1]$, while the rating in HRS, denoted as hrs_u, is a discrete number from 0 to 4, it is unable to match them directly. We evaluate the relevance according to the pairwise accuracy based on the number of concordances and discordances in preference pairs. Given two documents u_i and u_j under the same query, the concordant pair is that if $hrs_{u_i} > hrs_{u_j}$ and $rel_{u_i} > rel_{u_j}$, or if $hrs_{u_i} < hrs_{u_j}$ and $rel_{u_i} < rel_{u_j}$. An discordant pair is that if $hrs_{u_i} > hrs_{u_j}$ and $rel_{u_i} < rel_{u_j}$, or if $hrs_{u_i} < hrs_{u_j}$ and $rel_{u_i} > rel_{u_j}$. This pairwise accuracy is calculated as follows:

$$acc = 1 - \frac{D}{N} \tag{33}$$

Here, D represents the number of discordant pairs and N represents the total number of pairs generated by the click model.

Similarly, we compute the document relevance from the DBN and the CCM model according to the probability $P(C = 1|E = 1)$. After training click model, we generate the preference pair with respect to each pair of documents under the same query. However, we notice that the number of generated preference pairs from different click models varies significantly different. Thus, even one algorithm reaches better accuracy than another one, since the number of preference pairs is different, we cannot conclude which algorithm is better. In order to provide a fair evaluation, we introduce a threshold θ such that the preference pair $u_i > u_j$ is generated only when

$$rel_{u_i} - rel_{u_j} > \theta, \tag{34}$$

where $\theta \geq 0$. Thus, we can generate different set of preference pairs through setting different θ value. When we set θ as a larger value, less number of preference pairs are generated. Moreover, since the relevance difference becomes large, the generated preference pairs are more reliable. Accordingly, we evaluate the pairwise accuracy among different algorithms in terms of the similar number of preference pairs.

Figure 3 reports the result of pairwise accuracies among three click models. For each click model, we set a series of θ values to generate different number of preference pairs and compute related pairwise accuracies. As θ increases, the number of pairs decreases and the pairwise accuracy increases correspondly. When the pair number is 1 million, the PCC model reaches to the pairwise accuracy 82.8% while DBN and CCM reaches to 81.7% and 78.2% respectively. When the number of pairs is 0.5 million, PCC reaches to the accuracy 86.3% while DBN and CCM reaches to 83.9% and 78.6% respectively. On average, the PCC model achieves 2% and 5% accuracy improvement than that of the DBN and CCM models.

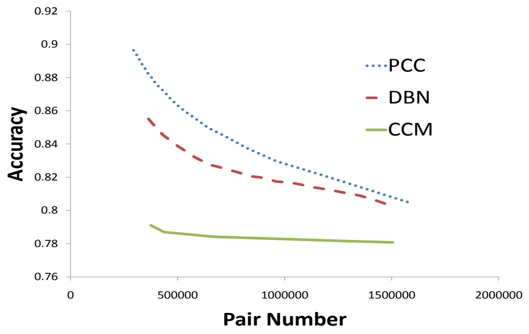

Figure 3: The pairwise accuracy comparison among three click models in terms of the number of preference pairs.

4.3 Ranking by Predicted Relevance

In the part, we use the predicted relevance to rank the documents directly. For one query and their related documents, every document is treated equally in computing the pairwise accuracy in the above. However, the ranking evaluation such as NDCG often put more emphasis on the documents at top positions. As such, the relative order of the documents with higher predicted relevance is more important than the documents with lower relevance.

For each query, we rank the returned documents according to the relevance value rel_{u_i} $(\forall i)$ and compute NDCG@1 and NDCG@3 scores for the PCC, DBN and CCM models. The results are shown in Figure 4 and 5, where we decompose the NDCG score in terms of query frequency. We can see when the query frequency is between 100 to 1000, NDCG@1 of the PCC model is 63.1%, which has 3% and 17% improvement than that of DBN and CCM, respectively. For extremely low frequent queries, the NGCG@1 improvement of the PCC model over DBN and CCM becomes less significant. The main reason is because the post-click features cannot be extracted for these queries and their related documents so that the post-click behaviors cannot contribute to the click model, which proves the effectiveness of incorporating post-click behavior into click model.

The overall NDCG@1 for all queries is 63.2%, which has 2% and 13% improvement over DBN and CCM. We also observe very similar results in NDCG@3, which demonstrates that the relevance inferred from PCC is consistently better than that from DBN and CCM.

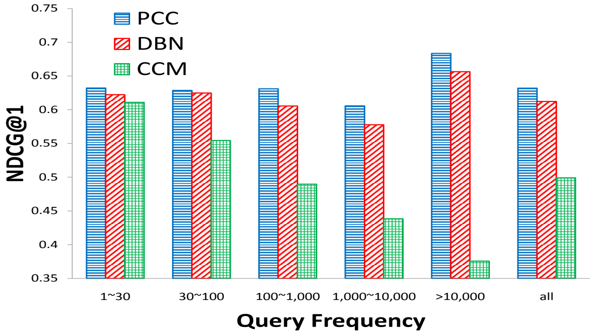

Figure 4: The NDCG@1 comparison among three click models in terms of query frequency.

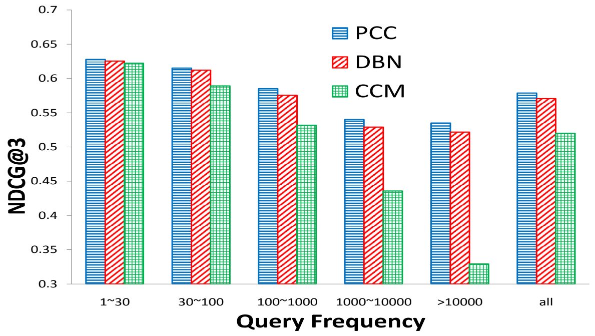

Figure 5: The NDCG@3 comparison among three click models in terms of query frequency

4.4 Integrating Predicted Relevance and HRS

Learning to rank is to optimize a ranking function from a set of documents with relevance ratings. We follow the RankNet [5] method which is a pairwise ranking algorithm receiving the pairwise preferences to optimize the ranking function. For each query and document, we extract about three hundred of features in the experiment, where the features are similar to those defined in LETOR[16]). Since the document relevance inferred from the PCC and DBN models is better than that from the CCM model in the above two experiments, we only consider the PCC and DBN models in this part of experiment.

We partition the HRS data as described in Table 2 into the training and testing sets. We randomly choose 3,000 queries and related 85,173 documents into the training set, and other queries and documents are in the testing data. There are totally about 5.1 million preference pairs generated from HRS as the training data. In addition, the click model are trained on the click log as described in Table 1, thus, there are about 7.4 million preference pairs generated from the PCC and the DBN. We construct three training sets for the RankNet: 1. only HRS; 2. PCC + HRS; 3. DBN + HRS, and evaluate the ranking function on the HRS testing data. The results on NDCG@1 and NDCG@3 are shown in Figure 6 and 7.

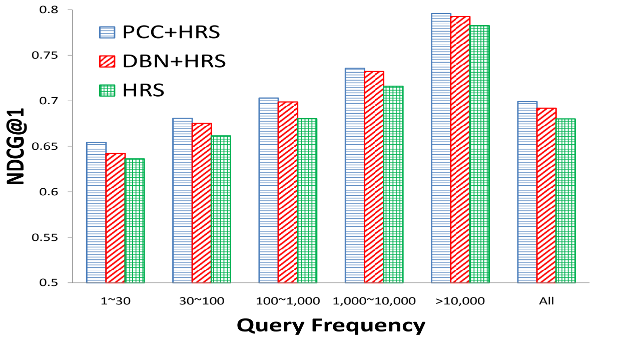

Figure 6: The NDCG@1 results from the RankNet algorithm on three different training sets.

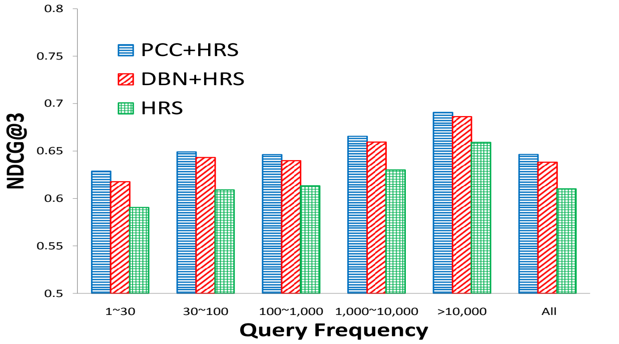

Figure 7: The NDCG@3 results from the RankNet algorithm on three different training sets.

The NDCG@1 and NDCG@3 results illustrate that the ranking function trained on the “PCC + HRS” data consistently outperform the function on the “DBN + HRS” data, while the function on the “DBN + HRS” data outperforms the function trained only on the “HRS” data. The overall NDCG@1 from “PCC+HRS” is 1.9% higher than that from “HRS”, which is a significant improvment of the ranking function on such large scale training and evaluation data.

4.5 Click Perplexity

Click perplexity is used as an evaluation metric to evaluate the accuracy of the click-through rate prediction. We assume that q_i^j is the probability of click drived from the click model, i.e. $P(C_i = 1|E_i = 1)$ at the position i and C_i^j is a binary value indicating the click event at the position i on the jth session. Thus, the click perplexity at the position i is computed as follows:

$$p_i = 2^{-\frac{1}{N}\sum_{n=1}^{N}(C_i^n log_2 q_i^n + (1-C_i^n) log_2 (1-q_i^n))} \qquad (35)$$

Thus, a smaller perplexity value indicates a better prediction.

The result on click perplexity is shown in Figure 8. We can see that the PCC model performs the best for the clicks in the first position. As for the other positions, the click perplexity from PCC are very similar to that from CCM. Although CCM has not inferred the document relevance very well in the above experiment, its click perplexity performs as well as PCC. The click perplexity obtained from PCC significantly outperforms the perplexity from DBN, which indicates that incorporating post-click behaviors into a click model can also produce a much better click prediction.

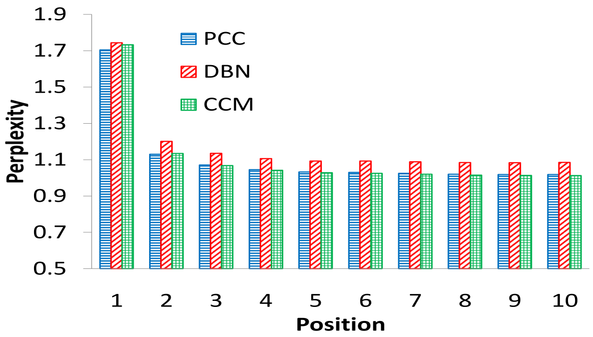

Figure 8: The click perplexity comparisons among three click models in terms of search position.

5. CONCLUSION AND EXTENSION

Besides user behaviors on the search result page, post-click behaviors after leaving the search page encodes very valuable user preference information. Different from previous works, this paper firstly investigates how to incorporate post-click behaviors into a click model to infer the document relevance. It proposes a novel PCC model by leveraging both click behaviors and post-click behaviors to estimate the degree of user satisfaction via a Bayesian approach. We conduct extensive experiments on a large scale data set and compare the PCC model with the state of the art works such as DBN and CCM. The experimental results show that PCC can consistently outperform baselines models on four different experimental setting. It is worth noting that the update of the PCC model is in a close form, which is capable of processing very large scale data sequentially.

The proposed method of incorporating post-click behaviors in the paper is a very general solution and can be extended to other click models such as CCM, UBM, etc. In the PCC model, the post-click behaviors are used as the features to estimate the user satisfication on the clicked document. However, it is not the only approach of incorporating post-click behaviors into click model. Another possible approach is to simulate user post-click behaviors through constructing a separate user browse model and then integrate it with the click models. We will explore these directions in future works.

6. REFERENCES

[1] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *proceedings of SIGIR2006*, 2006.

[2] E. Agichtein, E. Brill, and D. Susan. Improving web search ranking by incorporating user behavior information. In *proceedings of SIGIR2006*, 2006.

[3] R. Agrawal, A. Halverson, K. Kenthapadi, N. Mishra, and P. Tsaparas. Generating labels from clicks. In *proceedings of WSDM2009*, 2009.

[4] J. Attenberg, S. Pandey, and T. Suel. Modeling and predicting user behavior in sponsored search. In *proceedings of KDD2009*, 2009.

[5] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *proceedings of ICML2005*, 2005.

[6] B. Carterette and R. Jones. Evaluating search engines by modeling the relationship between relevance and clicks. In *proceedings of NIPS20*, 2008.

[7] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *proceedings of WWW2009*, 2009.

[8] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *proceedings of WSDM2008*, 2008.

[9] G. Dupret and B. Piwowarski. User browsing model to predict search engine click data from past observations. In *proceedings of SIGIR2008*, 2008.

[10] Y. Freund, R. Iyer, R.E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *The Journal of Machine Learning Research*, 4:933–969, 2003.

[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 WWW2009*, 2009.

[12] K. Jarvelin and J. Kekalainen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems 20(4), 422-446 (2002)*, 2002.

[13] T. Joachims. Optimizing search engines using clickthrough data. In *proceedings of KDD2002*, 2002.

[14] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *proceedings of SIGIR2005*, 2005.

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

[16] T.-Y. Liu, T. Qin, J. Xu, X. Wenying, and H. Li. Letor: A benchmark collection for research on learning to rank for information retrieval. *http://research.microsoft.com/en-us/um/beijing/projects/letor/*.

[17] T. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

[18] F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. In *proceedings of KDD2005*, 2005.

[19] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *proceedings of WWW2007*, 2007.

[20] D. Sculley, R.G. Malkin, S. Basu, and R.J. Bayardo. Predicting bounce rates in sponsored search advertisements. In *proceedings of KDD2009*, 2009.

7. APPENDIX

Since the computation of the coefficients ω_1, ω_2 and ω_3 is rather complicated, we move their equations into this section:

$$\omega_{1,k} = 1 - \frac{\lambda g(k-1,0)}{(1-\lambda)\sum_{j=0}^{k-2} g(j,0) + g(k-1,0)}$$

$$\omega_2 = (1-\lambda)\sum_{j=l}^{M-1} g(j,l) + g(M,l)$$

$$\omega_{3,k} = 1 -$$

$$\frac{\lambda P(S_{u_l}=0) g(k-1,l)}{P(S_{u_l}=1) + P(S_{u_l}=0)\Big((1-\lambda)\sum_{j=l}^{k-2} g(j,l) + g(k-1,l)\Big)}$$

where

$$g(i,j) = \begin{cases} \lambda^{i-j} P(A_{j+1}=0) \times \cdots \times P(A_i=0) & i > j \\ 1 & i \leqslant j \end{cases}$$