Exploring Relevance for Clicks[1]

Rongwei Cen, Yiqun Liu, Min Zhang, Bo Zhou, Liyun Ru, Shaoping Ma
State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology,
CS&T Department, Tsinghua University, Beijing, 100084, China P.R.
cenrongwei@gmail.com

ABSTRACT
Mining feedback information from user click-through data is an important issue for modern Web retrieval systems in terms of architecture analysis, performance evaluation and algorithm optimization. For commercial search engines, user click-through data contains useful information as well as large amount of inevitable noises. This paper proposes an approach to recognize reliable and meaningful user clicks (referred to as Relevant Clicks, RCs) in click-through data. By modeling user click-through behavior on search result lists, we propose several features to separate RCs from click noises. A learning algorithm is presented to estimate the quality of user clicks. Experimental results on large scale dataset show that: 1) our model effectively identifies RCs in noisy click-through data; 2) Different from previous click-through analysis efforts, our approach works well for both hot queries and long-tail queries.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process, Relevance feedback*

General Terms
Algorithms, Experimentation, Human Factors.

Keywords
Click-through Data, Click Relevance, Long-tail Query, Implicit Feedback.

1. INTRODUCTION
User feedback provides useful information for analyzing, evaluating and optimizing the performance of Web retrieval systems. Previous studies indicated that users are unwilling to provide explicit feedback for search engines [4]. Therefore, more studies (e.g. [1][3][5]) looked into implicit feedback information extracted from click-through data. This kind of feedback information has developed into an important research topic in the area of information retrieval and knowledge management, and has also been emphasized by commercial search engine community.

Unfortunately, practical Web data sources as well as click-through logs contain lots of noises. Individual users may behave irrationally or randomly, or may not even be real ones, and we cannot treat each user as an "expert" for labeling relevance [1]. By performing eye-tracking studies, Joachims et al. [5] showed that individual user clicks include bias and cannot be used directly as judgments of absolute relevance. Therefore, the study of the reliability and the relevance of clicks is an essential and fundamental work. To evade this click relevance problem, state-of-the-art approaches require a large volume of click-through data to extract user feedback information based on user group instead of individual user (e.g. [1][3][5]). The consequent problem is that these techniques only deal with hot queries with extensive user interaction data and are not applicable for long-tail ones.

Focused on this click relevance estimation problem, this paper proposes an approach to identify whether a click is reliable for labeling relevance. By analyzing search process of Web users, several features are proposed and a learning algorithm is derived to estimate relevance scores of clicks in a probabilistic notion. It is effective in dealing with queries with different frequencies, especially for long-tail queries.

2. RELATED WORK
In recent years, implicit feedback information has been receiving much attention in information retrieval area. Several approaches are proposed to mine relevant information from click-through data.

Tan et al. [9] detected robot behaviors for increasing the robustness of data. Baeza-Yates et al. [2] and Kammenhuber et al. [6] modeled user clicks, query formulations and pages visited and revealed several interesting aspects of user behavior. Sadagopan et al. [7] identified typical and atypical user sessions in click streams. Joachims et al. [5] analyzed users' decision processes in Web search using eye-tracking devices in a controlled, laboratory setting. In 2006, Agichtein et al. [1] aggregated information from many unreliable user search session traces instead of treating each user as an individual "expert". Dou et al. [3] studied the problem of using aggregate click-through logs, and found that the aggregation of a large number of user clicks provided a valuable indicator of relevance preference.

Most of above work found that although single user's click-through behavior was unreliable and biased, certain information could be extracted from user behavior information. Most of state-of-the-art methods applying click-through data are not based on individual users and clicks, which cannot be applicable for long-tail queries. In this paper, we propose a concept of Relevant Click (RC), analyze user click behaviors, and recognize Relevant Clicks from individual users at click level.

CIKM'09, November 2–6, 2009, Hong Kong, China.

[1] Supported by the Chinese National Key Foundation Research & Development Plan (2004CB318108), Natural Science Foundation (60621062, 60503064, 60736044) and National 863 High Technology Project (2006AA01Z141).

3. FORMAL SETUP

Before analyzing user click-through logs, we formalize the associations between queries and documents. Naturally, Bipartite Graph is applied to draw the associations. Given a bipartite graph $G = (Q, D, \Re)$, consisting of a query set Q, a set of document D, and an edge set $\Re$ representing the association between Q and D.

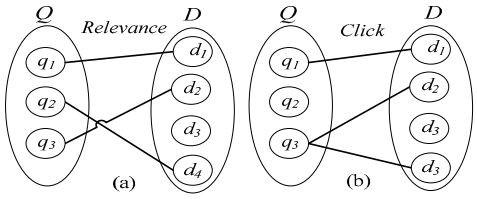

Figure 1. Sample relevance (a) and click (b) bipartite graphs of queries and documents.

Figure 1(a) illustrates the relevance bipartite graph connecting queries and documents and we define a binary relevance function *Rel(q,d)* over all documents $d \in D$ and all queries $q \in Q$ as follows:

$$Rel(q,d) = \begin{cases} 1, \text{if } q \text{ and } d \text{ are relevant} \\ 0, \text{if } q \text{ and } d \text{ are irrelevant} \end{cases}$$

Similarly, we define a binary click function *c(q,d)* over all documents $d \in D$ and all queries $q \in Q$ formally as follows:

$$c(q,d) = \begin{cases} 1, \text{if user clicked } d, \text{ querying } q \\ 0, \text{others} \end{cases}$$

Traditionally, the relevance function is constructed by human evaluation. Unfortunately, it is expensive and time-consuming, and annotators may fail to guess what real-world users are seeking for. To overcome this problem, many approaches were proposed to extract relevance information from click-through data. Due to vast noise and bias in the data, the click and the relevance function is not equal. Our task is to design a model that cannot only recognize a Relevant Click but also quantify the probability of one click being a Relevant Click.

Definition1: *Relevant Click (RC) is the click reflecting that the clicked document is relevant to the submitted search query.*

Formally, we describe *the set of Relevant Clicks* (*RCs*) as follows:

$$RCs \subseteq \{(q,d) \mid Rel(q,d) = 1,\ c(q,d)=1,\ \forall q \in Q,\ \forall d \in D\}$$

When *c(q,d)*=1, we abbreviate the click as *c* and the *Rel(q,d)* function as *Rel(c)*.

To estimate a click being relevant, we define the RC-Probability.

Definition2: *RC-Probability,* $\Re(c)$, *is the probability of a click c with a group of features* F_c *being a Relevant Click*:

$$\Re(c) \triangleq Pr[Rel(c) = 1 \mid F_c]$$

Actually, RC-Probability cannot only measure the probability that a click is relevant, but also be useful to rank clicks by their probability. That is, if we are given two clicks, c_1 and c_2, and c_1 has a lower $\Re(c)$ than c_2, then it should indicate that c_1 is less likely to be a RC than c_2. Such a probability function would at least be useful in ranking clicks.

4. FEATURES FOR CLICK RELEVANCE

Traditionally, user clicks are considered as a proof of relevance between queries and documents, and state-of-the-art approaches requires extensive user interaction data to guarantee statistical reliability. However, for long-tail queries, there is insufficient click data for statistical analysis. To assure our approach working for long-tail queries, we extracted features based on individual user clicks instead of relying on global statistics oriented features. Hence we look into user decision process, analyze user click behaviors, then observe and propose features for Relevant Clicks from individual users at click level.

When a user types a query into a search engine interface, and clicks some returned results, the results might satisfy user or not. According to the RC definition, a click is a RC if the user is satisfied by the clicked result. Being inspired by a lot of previous work on user click behavior [5] and search process analysis, we summarize the following evidences for RC:

- When a user clicks an irrelevant result, he is not satisfied and still need more information. Then the query tends to be refined and resubmitted, or more results will be clicked.
- For different positions in click sequence, user pays different attention to search results. Before clicking any result, user is likely to pay more attention to result list by comparing the information of each result, e.g. title, snippets, URL.
- User tends to stop when he finally reaches a satisfying document.

Based on the evidences above, we design a set of features as rich as possible that allow us to characterize whether a click satisfies a user or not. These features are general and query-independent, and they are not relied on global statistics oriented properties. The features applied to estimate clicks are summarized in Table 1.

Table 1. Features applied to represent user click behavior

Feature Name	Description
NumQueries	Unique query number in current session
ClickEntropy	Entropy of click distribution in current session
ClickSelfInformation	Self-information of current doc in current session
ClickDocNumInSession	Click number of current doc in current session
ClickDocNumInQuery	Click number of current doc in current query
IsFirstClickInSession	=1 if the first click in current session, =0 otherwise
IsLastClickInSession	=1 if the last click in current session,=0 otherwise
IsFirstClickInQuery	=1 if the first click in current query, =0 otherwise
IsLastClickInQuery	=1 if the last click in current query, =0 otherwise
ClickOrderInSession	Order of current click in current session
ClickOrderInQuery	Order of current click in current query
ClickDocRank	Result rank of current click

5. BAYESIAN LEARNING MODEL

Having described features of click behavior, we turn to construct RC-Probability $\Re(c)$. Our goal is to learn a general function automatically instead of relying on heuristic or intuitive insights. First, we review the RC-Probability $\Re(c)$.

If there is a click c with a group of features F_c, the RC-Probability is defined as follows:

$$\Re(c) \equiv Pr[Rel(c)=1 \mid F_c]$$

For the task of constructing RC-Probability, we consider two cases, the case where $\Re(c)$ is based on only one feature and the case where multiple features are involved.

Case 1: Single feature analysis. If we adopt only one user behavior feature F, $\Re(c)$ with F is denoted as $Pr[Rel(c)=1 \mid F]$ and we apply Bayesian theorem to rewrite this expression as:

$$Pr[Rel(c)=1 \mid F] = \frac{Pr[F \mid Rel(c)=1]}{Pr[F]} \times Pr[Rel(c)=1]$$

In equation above, $Pr[Rel(c)=1]$ is the proportion of relevant clicks in whole click-through data and it can be regarded as a constant value. So, we can rewrite the equation as:

$$Pr[Rel(c)=1 \mid F] \propto \frac{Pr[F \mid Rel(c)=1]}{Pr[F]}$$

Here, $Pr[F \mid Rel(c)=1]$ can be estimated using the proportion of F-featured clicks in the sample relevant click set. While $Pr[F]$ equals the proportion of F-featured clicks in all click-through data.

Case 2: Multiple feature analysis. If we use more than one user behavior feature, namely $F_c = \{F_1, \cdots, F_n\}$, and assume that each feature is conditionally independent of every other one, naïve Bayesian theorem assumes the following equation holds:

$$Pr[F_1, \cdots, F_n | Rel(c)=1] = \prod_i Pr[F_i | Rel(c)=1]$$

Similarly, we achieve that the following equation also holds approximately if each feature is independent of every other one:

$$Pr[F_1, \cdots, F_n] = \prod_i Pr[F_i]$$

Finally, we obtain:

$$\Re(c) \propto \prod_i \frac{Pr[F_i | Rel(c)=1]}{Pr[F_i]}$$

Here, we change the RC-Probability to a Bayesian model, and we annotate this model as *RC model* and the value of the model as *RC score*. According to the above derivation process, we know that the RC model have the same properties as RC-Probability and we can apply the RC model to estimate click relevance.

6. EXPERIMENTS AND RESULTS

6.1 Datasets

In order to analyze behavior patterns of Web retrieval users, we collected search engine access logs from Sep. 19, 2008 to Oct. 24, 2008, with the help of a commercial search engine company in China. The access logs contain over 194 million user clicks and 58.1 million user sessions. For evaluation we randomly sampled about 1700 queries from query logs and constructed an annotation set containing 10.03K relevant query-doc pairs.

This query set contains both hot and long-tail queries. In order to evaluate methods performance over different query sets with different frequencies, we created three subsets, Q1, Q2 and Q3. Table 2 shows the information of these data sets. From Table 2, we know that long-tail queries are the main part in click-through data, and 98.33% queries of whole unique ones are accessed less than or equal to 20 times.

Table 2. The information of datasets, Q1, Q2, Q3, and All

Query set	Q1	Q2	Q3	All
Query frequency	≤ 20	20~100	>100	Any
#(unique queries in set)	543	547	676	1766
#(individual click)	9390	51.18K	2.35M	2.41M
#(unique query in all logs)	18.0M	242.1K	63564	18.3M

6.2 Labeling Relevant Click

Now, we turned to experimental evaluation of predicting relevance of clicks. After learning, each click was assigned a RC score. We chose ROC (Receiver Operating Characteristics) curves and corresponding AUC (Area Under the ROC Curve) values to evaluate the performance of our method. ROC graphs [8] are commonly used in machine learning and data mining research, especially for quality estimation researches. ROC curves of RC model and random selection method are shown in Figure 2.

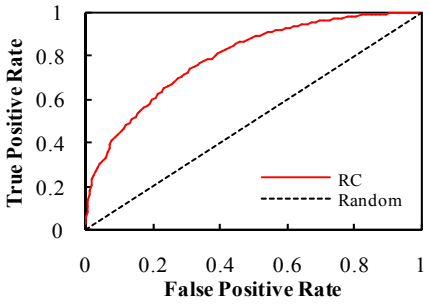

Figure 2. ROC curves to evaluate the performance of RC.

From Figure 6, we see that high scores were assigned to RCs, and it means that our method is able to select relevant clicks in a probabilistic notion and better than random selecting method. The AUC value for the RC algorithm's ROC curve is 0.792, meaning that our algorithm has 79.2% chances to rank a RC higher than a general click, while the AUC value for the random curve is 0.5.

The ROC curve shows that high reliable clicks are able to be selected using our algorithm probabilistically. Table 3 exhibits that when we filter out 80% less reliable clicks, the algorithm can maintain 60% relevant clicks and when we filter out 40% low reliable clicks, the algorithm maintain 92.8% relevant clicks.

Table 3. The retained RCs proportion with different threshold

Cleansed data size	20.0%	40.0%	60.0%	80.0%
Relevant click recall	60.0%	81.4%	92.8%	98.4%

6.3 Performance of Query Annotation

In this section, we apply the results of RC model to annotate relevance at query-doc pair level. Due to each query-doc pair clicked by amounts of users, there are a set of RC scores for each pair. The goal of annotation is to find an efficient method to label relevance of pairs. We compared two methods based on our RC model with traditional methods. The two methods are *Sum* and

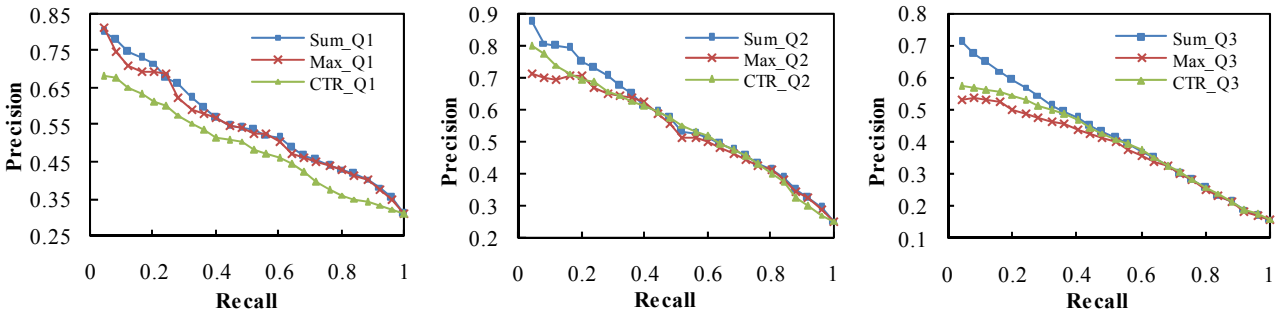

Figure 4. Precision vs. recall of RC and CTR methods over Q1,Q2 and Q3 datasets.

Max (a pair is relevant if the summation/maximum value of the score set is larger than a threshold) and *Max*.

We considered the effects of having sufficient or insufficient click logs available for a query and compared our RC methods (*Max* and *Sum*) to the traditional CTR method which rely on extensive user interaction data. Figure 4 reports recall-precision curves for the three methods for query sets with different frequencies, Q1, Q2, and Q3. The results indicate that the performance of our RC methods (Sum_QN, Max_QN, N=2,3) are closed to or better than CTR (CTR_QN,N=2,3) over the Q2 and Q3 sets whose queries have sufficient clicks. For the Q1 set, our method (Sum_Q1, Max_Q1) has better performance than CTR (CTR_Q1). It means that the CTR method fails for queries with insufficient clicks, while our method works well. Interestingly, Sum method exhibits higher precision than CTR at lower recall levels over the Q2 and Q3 sets.

To further validate the conclusion that our RC model is applicable for long-tail queries, two new query sets were constructed: Q4 contains queries with only one user access; Q5 contains queries with less than or equal to two user access. Then, we compared our methods to the CTR method over these two sets respectively. These two datasets can be viewed as long-tail queries absolutely.

The results present that our RC model does not rely on extensive user click data, while the CTR method does. Figure 5 reports recall-precision curves for these two methods over Q4 and Q5. This indicates that the traditional CTR method perform suboptimally in this setting, exhibiting precision 0.315 and 0.331 respectively over two datasets. This is because the CTR is a statistical method and it fails to mine useful information from rare clicks. In contrast, our RC model works well and exhibits higher precision than CTR method (larger than 0.5 at Recall of 0.4). It proves that our method is applicable for long-tail queries.

7. CONCLUSIONS

In this paper, we explore implicit feedback information from user click-through data. First, we formalize two associations of queries and documents, relevance and click, and propose RC-Probability to estimate the distance between these two associations. After that, 12 user behavior features are proposed. A Bayesian model is proposed to describe the RC-Probability by integrating behavior features. Experimental results show that our method can estimate the RC-Probability of individual user clicks successfully, and perform in annotating relevant clicks, especially for long-tail queries. Our paper is the first, to the best of our knowledge, to interpret user behavior based on individual user behavior, and succeed in processing long-tail queries for exploring relevant feedback information. The information mining can be applied to improve Web search ranking, evaluate search engine performance, detect click spam, etc.

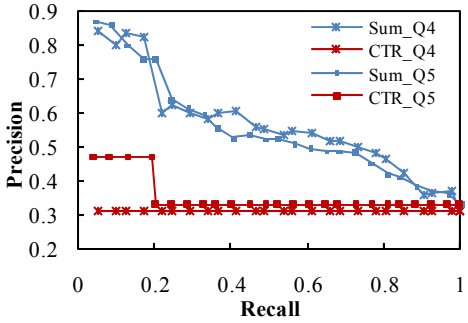

Figure 5. Precision vs. Recall of RC and CTR over Q4 & Q5.

8. REFERENCES

[1] Agichtein, E., Brill, E., Dumais, S., and Ragno, R. 2006. Learning user interaction models for predicting web search result preferences. In SIGIR '06. ACM, New York, NY, 3-10.

[2] Baeza-Yates, R., Hurtado, C., Mendoza, M., and Dupret, G. 2005. Modeling User Search Behavior. In LA-WEB 2005. IEEE Computer Society, Washington, DC, 242.

[3] Dou, Z., Song, R., Yuan, X., and Wen, J. 2008. Are click-through data adequate for learning web search rankings?. In CIKM '08. ACM, New York, NY, 73-82.

[4] Joachims, T., Freitag, D. and Michell, T. 1997. WebWatcher: a tour guide for the world wide Web. In IJCAI'97, Morgan Kaufmann, vol. 1, 770 – 777.

[5] Joachims, T., Granka, L., Pan, B., Hembrooke, H., Radlinski, F., and Gay, G. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in Web search. ACM Trans. Inf. Syst. 25, 2 (Apr. 2007), 7.

[6] Kammenhuber, N., Luxenburger, J., Feldmann, A., and Weikum, G. 2006. Web search clickstreams. In IMC '06. ACM, New York, NY, 245-250.

[7] Sadagopan, N. and Li, J. 2008. Characterizing typical and atypical user sessions in clickstreams. In WWW '08. ACM.

[8] Fawcett, T. 2006. An introduction to ROC analysis. Pattern Recogn. Lett. 27, 8 (Jun. 2006), 861-874.

[9] Tan, P. and Kumar, V. 2002. Discovery of Web Robot Sessions Based on their Navigational Patterns. Data Min. Knowl. Discov. 6, 1 (Jan. 2002), 9-35.