Content-Aware Click Modeling

Hongning Wang, ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana IL, 61801 USA
{wang296,czhai}@illinois.edu

Anlei Dong, Yi Chang
Yahoo! Labs
701 First Avenue, Sunnyvale, CA 94089
{anlei,yichang}@yahoo-inc.com

ABSTRACT

Click models aim at extracting intrinsic relevance of documents to queries from biased user clicks. One basic modeling assumption made in existing work is to treat such intrinsic relevance as an atomic query-document-specific parameter, which is solely estimated from historical clicks without using any content information about a document or relationship among the clicked/skipped documents under the same query. Due to this overly simplified assumption, existing click models can neither fully explore the information about a document's relevance quality nor make predictions of relevance for any unseen documents.

In this work, we proposed a novel Bayesian Sequential State model for modeling the user click behaviors, where the document content and dependencies among the sequential click events within a query are characterized by a set of descriptive features via a probabilistic graphical model. By applying the posterior regularized Expectation Maximization algorithm for parameter learning, we tailor the model to meet specific ranking-oriented properties, e.g., pairwise click preferences, so as to exploit richer information buried in the user clicks. Experiment results on a large set of real click logs demonstrate the effectiveness of the proposed model compared with several state-of-the-art click models.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval Models; H.3.5 [**Information Storage and Retrieval**]: Online Information Service

General Terms

Algorithms, Experimentation

Keywords

Click modeling, query log analysis, probabilistic graphical model

1. INTRODUCTION

User click logs provide rich and valuable implicit feedback information and can be used as a proxy for relevance judgments [14] or signals for directly influencing ranking [1]. However, they are also known to be vulnerable to position-bias – documents appearing at higher positions tend to receive more clicks even though they are not relevant to the query [10]. Therefore, properly modeling and interpreting the underlying mechanism that gives rise to user clicks is an important yet challenging research problem.

To fulfill this goal, click models [4, 5, 8, 11, 20] have been proposed for modeling user clicks and extracting intrinsic relevance information from the biased click logs. One fundamental assumption made in click models is the so-called *examination hypothesis*: a user clicks on a returned document *if and only if* that document has been examined by the user and it is relevant to the given query. Based on such an assumption, click models aim to distinguish the relevance-driven clicks from the position-driven clicks by postulating different dependency assumptions between the events of examining a document and clicking on it, e.g., position models [7, 8] and cascade models [4, 11]. Though deviating in various dependency assumptions about the examine and click events, all click models formalize a document's relevance quality to a given query as an *atomic* query-document-specific parameter, e.g., Bernoulli random variable [4, 11], which is solely estimated from multiple occurrences of such specific query-document pair in the click logs.

However, this commonly used modeling approach totally ignores the actual document content, which presumably would directly influence a user's click decision. As a result, the existing click models can neither take advantage of in-depth knowledge about the relevance quality of a document to the query buried in the document content, nor benefit from the semantic relation among the documents under the same query. For example, diversity is an important criterion for satisfying a user's information need [6]. A user would be less likely to click on a near-duplicate document if she has already clicked on a previous one with similar content; but in such a case, her "skip" decision does not necessarily mean that the document is irrelevant to her query. In most of the existing click models, we are only aware of which position is clicked, but the underlying "semantic explanations" for the clicking behavior, e.g., clicked content redundancy and click distance, are completely discarded.

A serious consequence of such an overly simplified assumption of a document's relevance quality to a given query is that the model's generalization capability is limited: one has to collect a large number of such query-document pairs to obtain a confident estimate of relevance. As shown in the previously reported results, only when there is a sufficient number of observations for the given query-document pairs,

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

could the existing click models demonstrate their advantages [4, 8, 11]. In the extreme case, when a new document comes into the search engine, there would be no way for us to accurately infer its relevance to the query immediately by a click model. The situation gets even severer in time-sensitive retrieval tasks, such as news search, where new documents keep emerging and we need timely estimation of their relevance quality to the given query before we could gather large number of user clicks.

In addition, existing click models only target at decomposing the relevance-driven clicks from the position-driven clicks, which boils down to discounting the observed clicks for each document in a *pointwise* manner. However, in a real search scenario, when a user decides to skip one document, it does not necessarily indicate the document is irrelevant to the query, since it is also possible that the previous/next clicked document is more relevant than it. Such property of user behavior has been proved by many real user studies [10, 14]. Therefore, existing click models are not optimized for distinguishing the relative order among the inferred relevance quality.

To the best of our knowledge, no existing work in click modeling attempted to address these two deficiencies, i.e., lack of exploring content information and failing to capture relative relevance preference. In this work, we propose to solve these limitations within a probabilistic generative framework, which naturally incorporates the document content and relative preferences between documents into click modeling. In detail, following the assumptions in cascade models, we propose a Bayesian Sequential State (BSS) model to formalize the generation of the observed clicks under a given query. First, to capture the rich semantic of a document's relevance quality to the query, we introduced a set of descriptive features (e.g., query matching in title and site authority) into query-document relevance modeling. Instead of hard coding the dependency among the click/examine events within a query (e.g., clicked documents must be relevant) [4, 11], we give our model the freedom to learn such relation from data based on the designed features, e.g., a click decision will be affected by the content redundancy between the current and previously clicked documents. Second, ranking-oriented knowledge, e.g., pairwise click preference, is incorporated by regularizing the posterior distribution of clicks, which helps us tailor the proposed probabilistic model and avoid undesirable local maxima.

The proposed model is a general click modeling framework, which covers most of existing models as special cases. On a large set of real click logs, the proposed BSS model outperformed several state-of-the-art click models in terms of relevance estimation quality. Especially when we only have limited size of training samples for a particular query-document pair, BSS model demonstrated its advantage by leveraging the information from ranking-oriented features for accurate relevance estimation. The introduced pairwise click preference renders BSS model better ranking capability in distinguishing the relative order of relevance among the candidate documents. Besides, BSS model provides a principled way of interpreting and modeling user's click behaviors, which is not available in existing click models.

2. BACKGROUND

The main purpose for modeling the user's click behaviors in search engine logs is to fight against the notorious position-bias and extract the document's intrinsic relevance to the query. Richardson et al. [19] attempted to combat position-bias by imposing a multiplicative factor on documents in lower positions to infer their true relevance. This idea was later formalized as the *examination hypothesis* and adopted in the position models [7]. The key assumption in position models is that the user clicks on a document *if and only if* that document has been examined by the user and it is relevant to the query. In addition, the examination event *only* depends on the position. Formally, given a document d displayed at position i, the probability of d being clicked (i.e., $C = 1$) is determined by the latent examination event (i.e., $E = 1$) as,

$$\begin{aligned}P(C=1|d,i)=&\sum_{e\in\{0,1\}}P(C=1|d,i,E=e)P(E=e|d,i)\\=&P(C=1|d,E=1)P(E=1|i)\end{aligned}$$

where $P(C = 1|d, E = 1)$ is specified by a document-specific parameter α_d describing the document's intrinsic relevance quality to the query, and $P(E = 1|i)$ is determined by a position-specific parameter β_i to capture position bias.

However, the pure position models deal with examination event in an isolated manner, i.e., the examination probability $P(E = 1|i)$ is assumed to be independent from the click events. Cascade models are one typical extension to conquer this limitation, which further assume the user will examine the returned documents from top to bottom and make click decisions over each examined document. Once the user stops examining, all the following documents will not be examined. Therefore, a click event in a query session is modeled as,

$$P(C_i=1)=P(R_i=1)\prod_{j=1}^{i-1}\big[1-P(R_j=1)\big]$$

where $R_i = 1$ is the event that document d at position i is relevant to the given query.

One drawback of the original cascade model is that it can only deal with queries containing one click, later work generalizes it to queries with multiple clicks. Chapelle et al., [4] solved this limitation by distinguishing the perceived and intrinsic relevance of a document: they assumed the perceived relevance controls the click event and the intrinsic relevance determines the user's satisfaction with the current document and her further examination of the following documents.

Our proposed BSS model falls into the category of cascade models: we assume the users would sequentially examine the returned documents from top to bottom for the given query, and a clicked document must be examined beforehand. In addition, by incorporating a set of ranking features, we model a document's relevance quality to a given query in a more general way: we assume the relevance quality of a document to the given query is not only an intrinsic property of the document itself, but also influenced by the displayed document content (e.g., title and abstract). The dependency relation between the examine and click events are flexibly learned from data, e.g., an examined and relevant document may still be skipped. In addition, the proposed method also explores the relationship among the clicked and skipped documents under the same query, e.g., content redundancy, which is not covered by existing click models. In previous work, click decision is only determined by the document's own relevance quality; while in our proposed model, ranking-oriented constraints, e.g., pairwise click preferences, are also incorporated to improve the model's capability of distinguishing the relative order of relevant documents.

3. BAYESIAN SEQUENTIAL STATE MODEL

As discussed earlier, existing click models have two limitations: 1) modeling the relevance of document to the given query as an *atomic* query-document-specific parameter; 2) failing to capture the relative order of estimated relevance between the documents. To break these two limitations and make click models applicable in more search scenarios, we propose a novel Bayesian Sequential State (BSS) model, in which the relevance quality of document to a given query is parameterized by a set of document-specific features, and the dependencies among the click and examine events within the same session are explicitly captured and exploited.

3.1 Basic Generative Assumption

Following the basic modeling assumption in *cascade models*, in our proposed BSS model, we assume that when a user submits a query to the search engine and gets a list of ranked results, she would sequentially examine the returned documents from top to bottom; a document must be examined before she clicks on it; and once she decides to stop examining at current position, she would leave this query session without further interactions. In particular, we assume that when she is examining a document, she would *judge* its relevance according to the displayed document content, e.g., title and abstract, which can be characterized by a set of features, e.g., query term matching in title and abstract; in addition, the user *remembers* her previously examined documents under this query, so that when she moves onto lower positions, her previous click/skip decisions will affect her later choices, e.g., skipping the less relevant documents. In other words, the click/skip events within the same query session are assumed to be dependent with each other.

Formally, assume there are N queries in our collection and for each query there are M ordered documents. Following the notations introduced in Section 2, we use binary variables to denote the relevance status, examine and click events of a document, i.e., $R = \{0, 1\}$, $E = \{0, 1\}$ and $C = \{0, 1\}$. To make the presentation concise, we will ignore the symbol d_i representing the document displayed at position i under a particular query, when no ambiguity is caused. Hence, the generation process of the observed clicks in a collection of query logs defined by the proposed BSS model can be formalized as follows:

- For each query q in the query log:
    - For document d in position i:
        1. Decide whether to examine the current position based on previous examination event E_{i-1} and previous document d_{i-1}'s relevance status R_{i-1}, i.e., $E_i \sim P(E_i|E_{i-1}, R_{i-1}, q)$. If $i = 1$, $E_i = 1$;
        2. If $E_i = 0$, abandon further examination;
        3. Judge d_i's relevance against query q, i.e., $R_i \sim P(R_i|d_i, q)$;
        4. Decide whether to click d_i based on its relevance quality, i.e., $C_i \sim P(C_i|E_i, R_i, q)$

As a result, the joint probability of random variables $\{E_i, R_i, C_i\}_{i=1}^M$ within a search result page for query q can be formulated as:

$$P(\boldsymbol{E}, \boldsymbol{R}, \boldsymbol{C}|q) = \prod_{i=1}^{M} P(C_i|R_i, E_i, q)P(E_i|R_{i-1}, E_{i-1}, q)P(R_i|d_i, q) \tag{1}$$

Different from most of the existing click models, where the dependency relation is hard-coded in their conditional probabilities, e.g., an examined and relevant document must be clicked: $E_i = 1, R_i = 1 \Leftrightarrow C_i = 1$ [4, 11], we relax such hard requirement to accommodate noise in clicks [5]. We assume that even an examined document is not relevant, the user might still click on it because of her carelessness, i.e., $P(C_i = 1|E_i = 1, R_i = 0) > 0$; and on the other hand, even if an examined document is relevant, the user might still skip it due to the redundancy or her satisfaction of pervious clicks, i.e., $P(C_i = 0|E_i = 1, R_i = 1) > 0$. In addition, to fully explore the dependency between a click event and the document's relevance status, we assume user's further examination also depends on the current document's relevance quality, i.e., $P(E_i|E_{i-1}, R_{i-1}) \neq P(E_i|E_{i-1})$.

3.2 Conditional Probability Refinement

The generation process introduced in Eq (1) depicts the skeleton of dependencies among the random variables of $\{E_i, R_i, C_i\}_{i=1}^M$ within the search result page for a given query. Next, we will discuss the details of how we can incorporate descriptive features to materialize those dependency relations and exploit rich information conveyed in the users' click behaviors.

To parameterize the dependency, we define the conditional probabilities in BSS model via logistic functions:

1. Relevance probability:

$$P(R_i = 1|d_i, q) = \sigma({w^R}^{\mathsf{T}} f^R_{q,d_i} + w^R_{q,d_i}) \tag{2}$$

2. Click probability:

$$P(C_i = 1|R_i, E_i, q) = \begin{cases} 0 & \text{if } E_i = 0 \\ \sigma({w^C_{R=0}}^{\mathsf{T}} f^C_{q,d_i}) & \text{if } E_i = 1, R_i = 0 \\ \sigma({w^C_{R=1}}^{\mathsf{T}} f^C_{q,d_i}) & \text{if } E_i = 1, R_i = 1 \end{cases} \tag{3}$$

3. Examine probability:

$$P(E_i = 1|R_{i-1}, E_{i-1}, q) = \begin{cases} 0 & \text{if } E_{i-1} = 0 \\ \sigma({w^E_{R=0}}^{\mathsf{T}} f^E_{q,d_i}) & \text{if } E_{i-1} = 1, R_{i-1} = 0 \\ \sigma({w^E_{R=1}}^{\mathsf{T}} f^E_{q,d_i}) & \text{if } E_{i-1} = 1, R_{i-1} = 1 \end{cases} \tag{4}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$, $\{f^R_{q,d}, f^C_{q,d}, f^E_{q,d}\}$ are the features characterizing the conditional probabilities for relevance status, click and examination events of document d_i under query q; and $\Theta = \{w^R, w^C_{R=0}, w^C_{R=1}, w^E_{R=0}, w^E_{R=0}\}$ are the corresponding importance weights for the features.

In particular, to distinguish the intrinsic relevance and perceived relevance, we assume a document's latent relevance status to a given query is determined by the mixture of these two types of relevance, i.e., $P(R_i = 1|q) = \sigma({w^R}^{\mathsf{T}} f^R_{q,d} + w^R_{q,d})$ as defined in Eq (2). In particular, $w^R_{q,d}$ is a scaler factor reflecting the intrinsic relevance quality of a document to the given query, which is assumed to be drawn from a zero mean Normal distribution. And ${w^R}^{\mathsf{T}} f^R_{q,d}$ is an

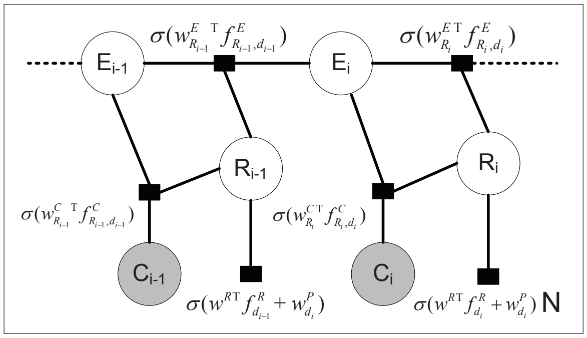

$\sigma(w_{R_{i-1}}^{E}{}^{\mathrm{T}} f_{R_{i-1},d_{i-1}}^{E})$
$\sigma(w_{R_i}^{E\mathrm{T}} f_{R_i,d_i}^{E})$
E_{i-1}
E_i
R_{i-1}
R_i
$\sigma(w_{R_{i-1}}^{C}{}^{\mathrm{T}} f_{R_{i-1},d_{i-1}}^{C})$
$\sigma(w_{R_i}^{C\mathrm{T}} f_{R_i,d_i}^{C})$
C_{i-1}
C_i
$\sigma(w^{R\mathrm{T}} f_{d_{i-1}}^{R} + w_{d_i}^{P})$
$\sigma(w^{R\mathrm{T}} f_{d_i}^{R} + w_{d_i}^{P})$
N

Figure 1: Factor graph representation for the proposed BSS model. Circles denote the random variables and black squares denote the conditional probabilities defined in Eq(2)-(4). Random variable E_i implies whether document d_i is examined, R_i represents d_i's relevance status to the query, and C_i indicates whether d_i is clicked by the user.

estimate of the perceived relevance quality, which is characterized by a weighted combination of relevance-driven features $f^R_{q,d}$, e.g., site authority and query term matching in document title. When we have sufficient observations of the query-document pair (q, d), the estimation of $w^R_{q,d}$ will be close to its true intrinsic relevance; but when we only have limited observations, e.g., for a new document, relevance-driven features $f^R_{q,d}$ will help to identify its perceived relevance, which leads to user clicks.

Using the language of probabilistic graphical models, we summarize the specification of the proposed BSS model by a factor graph representation in Figure 1.

3.3 Feature Instantiation for BSS

Table 1 lists the detailed definition of the proposed features in BSS model, which aim at capturing different factors affecting a user's click decision.

Among the proposed features, $f^R_{q,d}$ is the set of features describing the relevance quality of a document to the given query. This is the core problem for modern information retrieval study, and many effective features have been proposed for this purpose, such as BM25 and PageRank. In this work, we utilized 65 text matching features (e.g., query term matching in document title and abstract) as our relevance features. We should note that the proposed model is general and can potentially accommodate any combination of relevance-driven features.

Table 1: Features for materializing conditional probabilities in BSS model.

Type	Description	Value
f^R_{q,d_i}	65 text matching features e.g., query matching in title, query proximity in abstract	-
f^C_{q,d_i}	position	i
	# clicks	$\sum_{j<i} \mathbf{1}[C_j = 1]$
	distance to last click	$i - \arg\max_{j<i}[C_j = 1]$
	query length	$\|\|q\|\|$
	clicked content similarity	$\mathrm{AVG}_{j<i,C_j=1} sim(d_i, d_j)$
	skipped content similarity	$\mathrm{AVG}_{j<i,C_j=0}[sim(d_i, d_j)]$
f^E_{q,d_i}	position	i
	# clicks	$\sum_{j<i} \mathbf{1}[C_j = 1]$
	distance to last click	$i - \arg\max_{j<i} [C_j = 1]$
	avg content similarity	$\mathrm{AVG}_{j<i,k<i}[sim(d_j, d_k)]$
	variance content similarity	$\mathrm{VAR}_{j<i,k<i}[sim(d_j, d_k)]$

(All three types of features also include an additional bias term b accordingly.)

Though we are aiming to distinguish different effects of the current document's relevance status in examination and click events, it is impossible for us to pre-categorize which set of features would only affect user's click (examine) decision when the current document is relevant and vice versa. We decide to use the same set of features for these two situations, but give them different weights, i.e., $\{w^C_{R=0}, w^C_{R=1}\}$ for click event and $\{w^E_{R=0}, w^E_{R=1}\}$ for examine event, to portray their distinct contributions. In detail, the click-event-related features $f^C_{q,d}$ are used to indicate how the user would behave when an examined document is judged to be relevant ($R = 1$) or irrelevant ($R = 0$). For example, when the document is irrelevant, a mis-click might be caused by the position of the document (the user trusts more about the top ranked documents); and when the document is relevant, a skip decision may be due to the content redundancy of the clicked documents or her satisfaction of current search result (number of clicks). And the examine-event-related features $f^E_{q,d}$ exploit the factors affecting a user's examine decision on the next position. For example, when the current document is irrelevant ($R = 0$) and the user has skipped several documents in a row (e.g, distance to the last click), she would be more likely to give up further examining; and when the current document is relevant ($R = 1$) and the clicked documents are quite similar to each other so far (average content similarity), she might be more likely to stop.

3.4 Inference and Model Estimation

When applying the proposed BSS model in the testing phase, we do not need to restrict ourself to the documents ever occurred in the training set (i.e., $w^R_{q,d}$ exists). Since we have formalized the perceived relevance by a set of relevance-driven features, we can directly apply the model to any unseen document by calculating $\sigma({w^R}^{\mathsf{T}} f^R_{q,d})$ as an estimate of its relevance quality to the query (i.e., using mean value of the intrinsic relevance $w^R_{q,d}$ from prior for all the new candidate documents). And for those documents occurred in our training set, we can follow Eq (2) to incorporate the intrinsic relevance of document to the given query learned from the training set.

In model learning phase, because a document's relevance quality and examination status are not observed in the click logs, we appeal to the Expectation Maximization algorithm [18] to estimate the optimal parameter setting, which maximizes the lower bound of the log-likelihood of the observed click events in the training set,

$$L(\mathbf{C}, \mathbf{q}, \Theta) = \sum_{q,i} \log \sum_{E_i,R_i} p(E_i, R_i, C_i|q, \Theta)$$
$$\geq \sum_{q,i} \sum_{E_i,R_i} p(E_i, R_i|C_i, q, \Theta) \log p(E_i, R_i, C_i|q, \Theta) \quad (5)$$

Particularly, in E-Step, we calculate the posterior distribution of $P(\boldsymbol{E}, \boldsymbol{R}|\boldsymbol{C}, q, \Theta^{(t)})$ for the latent variables $(E_i, R_i)_{i=1}^M$

in a query session with respect to the current model $\Theta^{(t)}$. One advantage of the proposed model is that, in the model training phase, since the clicked documents are already known, we can fix them and reduce the maximum clique size in the induced graph structure to 3, i.e., $\{R_{i-1}, E_{i-1}, E_i\}$. As a result, exact inference is tractable and can be efficiently calculated via Belief Propagation [15]. And in M-Step, we obtain the new model parameter $\Theta^{(t+1)}$ by maximizing the expectation of the "complete" log-likelihood under $P(\boldsymbol{E}, \boldsymbol{R}|\boldsymbol{C}, q, \Theta^{(t)})$ as defined in Eq (5), which can be solved by any standard optimization technique (in this work, we used L-BFGS [17]). The E-Step and M-Step are alternatively executed until the relative change of the righthand side of Eq (5) is smaller than a threshold.

3.5 Discussion

There are close connections and clear differences between the proposed BSS model and other existing click models. First, BSS model explicitly encodes a document's relevance quality to a given query as a mix of intrinsic relevance and perceived relevance, which makes it feasible to incorporate richer information conveyed in document content for relevance estimation. Second, BSS model generalizes the dependency between a click event and the corresponding document's examine and relevance status. Most of previous work puts hard constraint over the click event, i.e., $C_i = 1 \Leftrightarrow E_i = 1, R_i = 1$, which fails to recognize noisy clicks and dependency among documents under the same query. Third, the conditional probabilities defined in BSS are no longer simply treated as document- or position-specific parameters; instead, a set of descriptive features are designed to capture rich semantics of users' click behaviors.

If we resume the hard dependency setting and drop most of the newly introduced features, the proposed BSS model can be easily adopted to many existing click models: the examination model proposed in [19] can be treated as a special case of our BSS model if we remove all the examine features except position and assume it is independent of previous relevance status, i.e., $w^E_{R=0} = w^E_{R=1}$. And if we disable the relevance features $f^R_{q,d}$ and only keep $w^R_{q,d}$ for each query-document pair in the logistic function, we will go back to the traditional setting for the click models. Based on this, if we further remove the examination and click features, it reduces to the CCM model proposed in [11]; if we only keep the examine feature of *distance to last click*, it will reduce to the UBM model proposed in [8]; and if we restrict the examine probability to be $E_i = 1 - R_{i-1}$, it will reduce to the original cascade model [7], since the user has to keep examining until the first click.

From the above discussion, we can clearly notice that the proposed BSS model is a more general framework for modeling users' click behaviors: through parameterizations, many informative signals and dependency relation are introduced to help the model explore a document's in-depth relevance quality to the given query from historic clicks.

4. POSTERIOR REGULARIZATION

One potential problem of the current BSS model setting is that the designed structure is too flexible for the learning procedure to identify the "true" parameters, which depict the underlying dependency among the latent variables. One obvious deficiency is that a document's relevance status, $R_i = 1$ or $R_i = 0$, is interchangeable. Since the click/examine events are determined by the same set of features (weights to be learned from data), if we switch the labels of R_i in the whole collection, the model will find another optimal weight setting (switch the weights) to maximize the likelihood, but that is undesirable.

The main reason for this unidentifiable problem is that to capture noise within the click events we did not set hard constraints on the conditional probability of click events, i.e., we allow $P(C_i = 1|E_i = 1, R_i = 0) > 0$ and $P(C_i = 1|E_i = 1, R_i = 1) < 1$; but it gives too much freedom to these two conditional probabilities, such that they can freely exchange their roles and still maximize the likelihood of clicks. Existing click models avoid this unidentifiable problem by hard-coding the click events, i.e., $C_i = 1 \Leftrightarrow E_i = 1, R_i = 1$. In our work, to keep the flexibility of the modeling assumptions and handle the noisy clicks, we decide to regularize the posterior distribution inferred by the model.

Another benefit of posterior regularization is that we can easily incorporate the ranking-oriented knowledge, i.e., pairwise preference, into click modeling, which is hard to be directly encoded in the original conditional probabilities.

4.1 Posterior Regularized EM Algorithm

Posterior Regularization (PR) proposed by Ganchev et al. [9] is a general framework for postulating structural constraints over the latent variable models. The method roots in the block coordinate ascent EM framework [18], and it modifies the E-step of a standard EM algorithm to inject constraints over the posterior distribution of latent variables via the form of expectations. And such regularization will not affect the convergency of original EM algorithm. Taking our problem as an example, we should expect that the number of relevance-driven clicks should be larger than mistaken clicks, e.g., $E[C = 1, E = 1, R = 1] > E[C = 1, E = 1, R = 0]$.

Formally, the regularized E-step in PR framework aims to optimize:

$$\min_{q,\xi} \quad KL(q(Y)||p(Y|X, \Theta^{(t)})) \tag{6}$$

$$s.t.\ E_q[\phi(X,Y)] - \mathbf{b} \le \xi \tag{7}$$

$$||\xi||_\beta \le \epsilon$$

where $p(Y|X, \Theta^{(t)})$ is the original posterior distribution of the latent variables Y given the current model $\Theta^{(t)}$ and observation X, $q(Y)$ is the regularized posterior distribution of Y, $\phi(X, Y)$ is the constraint function defined over (X, Y), and ξ is a slack variable to relax the constraints. In our case, $Y = \{E_i, R_i\}^M_{i=1}$ and $X = \{C_i\}^M_{i=1}$

The convenience of PR framework comes from its dual form: the primal solution $q^*(Y)$ is uniquely determined in terms of the dual solution λ^* by,

$$q^*(Y) = \frac{p_\Theta(Y|X)\exp\{-\lambda^*\phi(X,Y)\}}{Z(\lambda^*)} \tag{8}$$

and the dual problem is defined as,

$$\max_{\lambda \ge 0} -\mathbf{b}^T\lambda - \log Z(\lambda) - \epsilon||\lambda||_{\beta^*} \tag{9}$$

where $Z(\lambda)$ is the partition function for Eq(8), and $||\lambda||_{\beta^*}$ is the dual norm of $||\lambda||_\beta$.

Eq (9) can be solved by the projected gradient algorithm [3], and Eq (8) can be effectively computed via Belief Propagation algorithm by factorizing the constraints according to the original factor graph. Intuitively, the PR framework can be thought as regularizing the posterior inference in E-Step of the original EM algorithm, such that the posterior distribution of the latent variables could satisfy some desired properties specified in the expectations.

4.2 Constraints for Posterior Regularization

In this section, we discuss the constraint that we designed to conquer the unidentifiable problem and that to incorporate the search-oriented pairwise constraints into our BSS model. In detail, we choose to relax the posterior constraints by setting ϵ to be a small constant (0.01), and use L2-norm to regularize the slack ξ.

4.2.1 *Dampen noisy clicks*

As we have discussed before, we need to restrict the influence of the noisy clicks, and we hypothesize that most of the clicks are driven by the relevance quality of the corresponding document. To achieve this, we define the constraint over the click events as:

$$\phi_{noise}(X,Y) = \sum_i \phi_{noise}(X, Y_i) \tag{10}$$

$$= \sum_i \begin{cases} -1 & \text{if } E_i = 1 \text{ and } R_i = C_i \\ c & \text{if } E_i = 1 \text{ and } R_i \neq C_i \\ 0 & \text{otherwise} \end{cases}$$

and set the left-hand side constant $\mathbf{b}$ to be zero in Eq(6).

The meaning of this constraint is straightforward: we require the ratio between the expectation of relevance-driven clicks ($E_i = 1, R_i = C_i$) and noisy clicks ($E_i = 1, R_i \neq C_i$) under the same query to be below a constant c, i.e., $\mathbf{E}[E_i = 1, R_i = C_i] > c\mathbf{E}[E_i = 1, R_i \neq C_i]$. In other words, we require at least $\frac{c}{c+1}$ clicks should be explained by the relevance quality of the document rather than a mistake.

4.2.2 *Reduce mis-ordered pairs*

Pairwise click preference can be easily incorporated via the PR framework. In this work, we encoded two frequently employed click heuristics, i.e., *skip above* and *skip next* [14], by the constraints defined below:

$$\phi_{pair}(X,Y) = \sum_i \phi_{pair}(X, Y_i) \tag{11}$$

$$= \sum_i \begin{cases} 0 & \text{if } E_i = 0 \\ 0 & \text{if } E_i = 1, C_i = 1 - C_{i-1}, R_i = C_i, R_{i-1} = C_{i-1} \\ 1 & \textit{otherwise} \end{cases}$$

and set the left-hand side constant $\mathbf{b}$ to be 0 in Eq(6).

The meaning of this constraint is: we only put constraint over the examined documents (i.e., $E_i = 1$) where the user makes different decisions in the adjacent positions (i.e., *skip above* and *skip next*). If the inferred relevance is consistent with the observed click preference (i.e., $R_i = C_i$ and $R_{i-1} = C_{i-1}$), such a constraint is inactive; otherwise, if the inferred relevance preference contradicts the observed click preference, we need to penalize it.

5. EXPERIMENT RESULTS

As we have discussed most of existing click models treat the relevance quality of a document to the given query as a static property, and therefore the evaluation is mostly performed in general web search logs, where the relevance quality of a document to a query is relatively stable. In this work, we are more interested in evaluating the effectiveness of the click models in a more dynamic search environment, i.e., news search, where new documents keep emerging, and existing documents quickly become out-of-date and fall out of the top ranked results. In such a scenario, we cannot expect to collect a large number of clicks for each document before we can make a confident relevance estimation.

5.1 Data Sets

We collected a large set of real user search logs from Yahoo! news search engine[1] in a two months period, from late May to late July 2011. During this period, a subset of queries are randomly selected and all the associated users' search activities are collected, including the anonymized user ID, query string, timestamp, top 10 returned URL sets and the corresponding user clicks. In order to unbiasedly compare the relevance estimation performance among different click modeling approaches, we also set up a random bucket to collect exploration clicks from a small portion of traffic at the same time. In this random bucket, the top four URLs were randomly shuffled and displayed to the real users. By doing such random shuffling, we were able to reduce the noise from position-bias in the collected user click feedback, and such feedback can be used as a reliable proxy on information utility of documents [16]. Therefore, we only collected the top 4 URLs from this random bucket. In addition, we also asked editors to annotate one day's query log on Aug 9, 2011, into five-level relevance labels, e.g., "Bad", "Fair", "Good", "Excellent" and "Perfect", immediately one day after to ensure the annotation quality.

Simple pre-processing is applied on these click data sets: 1) filter out the queries without clicks in the random bucket, since they are useless for testing purpose; 2) discard the queries only appearing once in the whole collection; 3) normalizing the relevance features $f^R_{q,d}$ by their mean and variance estimated on normal click set, i.e., z-score [21]. After these pre-processing steps, we collected 460k queries from the normal click set and 378k queries from the random bucket set. One thing we should note is that because of the way we set up the random bucket, many queries and documents might only appear in the random bucket. Existing click models can hardly estimate the relevance quality of such unseen documents. In order to make a comprehensive comparison, we split the normal click set into two subsets, and ensure each query is evenly distributed in these two subsets. We choose one of them for training purpose and another for testing. The basic statistics of the four data sets used in our experiment are listed in Table 2.

Table 2: Statistics of evaluation corpus.

	# Unique Query	# Query
Normal training clicks	11,701	234,149
Normal testing clicks	6,264	225,452
Random bucket clicks	33,762	378,403
Editorial judgment	1,404	13,091

In order to test the model's generalization capacity, we further split the queries in the normal click testing set and random bucket click testing set into different categories according to their frequencies in the training set. The basic statistics of those categories are shown in Figure 2. As can be clearly noticed in the figure, a large portion of testing

[1] `http://news.search.yahoo.com/`

queries in the random bucket set belong to the less frequent query category (62.92% queries are in the <25 category) comparing to the normal click set (11.49%), which makes the prediction more difficult in the random bucket set.

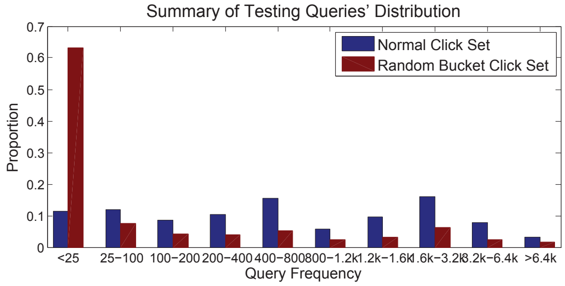


Figure 2: Distribution of testing queries according to their frequencies in training set.

5.2 Quality of Relevance Modeling

The main question to be answered in our experiments is whether the proposed model is more accurate than the existing click models in terms of relevance estimation. To answer this question and evaluate the quality of relevance modeling of the proposed BSS model, we compared it with a set of state-of-the-art click models, including the counting-based models of Dynamic Bayesian Model (DBM) [4] and User Browsing Model (UBM) [8], and feature-based models of Logistic Regression model and Examination Model [19]. Among them, Logistic Regression model and Examination Model are trained on the same set of 65 relevance features $f^R_{q,d}$ as our BSS model.

5.2.1 Evaluation metrics

In previous work [8, 11, 20], perplexity on the testing click set was often used as the metric for comparing different click models, and it is defined as,

$$2^{-\frac{1}{N}\sum_{i=1}^{N}\delta(c_i=1)\log_2 p(c_i=1)+\delta(c_i=0)\log_2 p(c_i=0)}$$

where N is the number of observations in the testing set. The lower perplexity a model can achieve, the closer its prediction is to the observation in the testing set.

However, such evaluation metric is problematic for two major reasons. First, clicks in the testing query log is still position-biased: a less relevant document appears at a higher position would still receive more clicks, such that a model correctly downgrades such a document will even get penalized by the perplexity metric. Second, since perplexity is defined based on the absolute value of the predicted probabilities, it is inherently sensitive to scaling or normalization of these probabilities, making it difficult to interpret the results appropriately.

To examine whether these two concerns are empirically supported, we included a simple baseline for click modeling, Naive Click Model (NCM), which only uses the frequency of clicks on a particular query-document pair observed in the training set as its relevance estimation. In Table 3, we compared NCM's perplexity against other sophisticated click models on normal click testing set. And to compare their relevance estimation quality, we also evaluated their P@1 ranking performance on the random bucket click set, which is proved to be an unbiased proxy of document's relevance quality [16]. Due to space limitation, we did not show the result from Logistic Regress Model.

Table 3: Comparison between perplexity metric and ranking metric.

	Examine	UBM	DBN	BSS	NCM
perplexity	3.9534	1.5471	1.5719	1.8213	**1.2925**
P@1	0.3807	0.3603	0.3494	**0.4033**	0.3578

From Table 3, we can clearly notice that the naive baseline outperforms all the other click models in perplexity on the normal click testing set, but its ranking performance is not the best on the unbiased random bucket click set. Besides, we also observed that the perplexity of Examination Model is significantly larger than the other click models. We looked into the detailed output of Examination Model and found that its predicted click probabilities (with mean 0.78) are much larger than the other models' predictions. Since the probability of a document being clicked in the normal click testing set is generally small (with mean 0.14), Examination Model get seriously penalized by perplexity. However, Examination Model's ranking performance is much better than NCM in the random bucket click set. We thus conclude that the perplexity calculated based on position-biased clicks is not a trustable metric for measuring a click model's capacity of recognizing relevant documents.

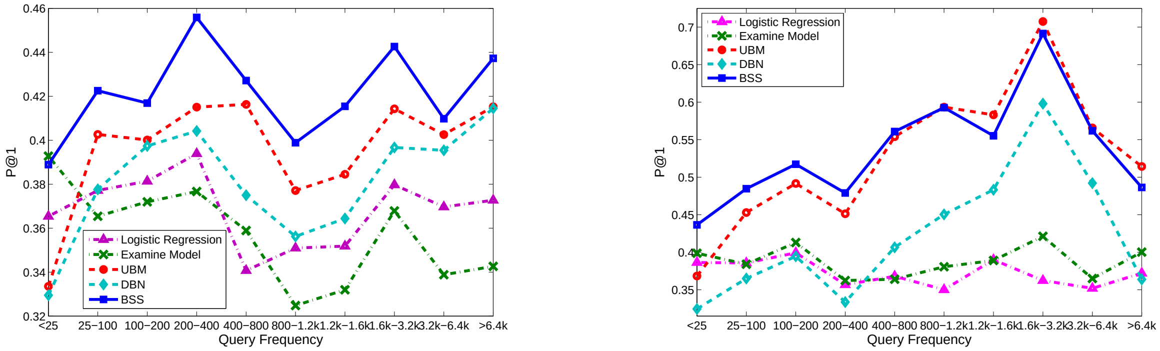
A potentially better measure than perplexity is to directly compare different click models' ranking performance based on the estimated relevance of documents. To evaluate ranking performance in a click-based data set, we treat all the clicked documents as relevant and calculate the corresponding *Precision at 1* (P@1), *Precision at 2* (P@2), *Mean Average Precision* (MAP) and *Mean Reciprocal Rank* (MRR). Definitions of these metrics can be found in standard textbooks in information retrieval (e.g., [2]). And in the editorial annotation data set, we treated the grade "Good" and above as relevant for precision-based metrics, and also included the normalized discounted cumulative gain (NDCG) [12] as an evaluation metric. Compared with perplexity metric, such a ranking-based evaluation can better reflect the utility of a click model in estimating the relevance of a document.

We evaluate the quality of relevance estimation of these models from two different perspectives: one is to directly use the estimated relevance from the click models to rank the documents; and another is to treat such relevance estimation as signals for training a learning-to-rank algorithm.

5.2.2 Estimated relevance for ranking

In this approach of evaluation, we ranked the candidate documents with respect to the estimated relevance given by a click model, and compared the ranking result against the logged user clicks. The higher position a click model can put a clicked document on, the better ranking capability it has. We performed the comparison on both random bucket click set and normal testing click set.

We first compared different models' P@1 performance on the random bucket click set in Figure 3 (a), where we illustrated the detailed comparison results under each category of different query frequencies. Li et al. [16] proved that P@1 metric on this random bucket click set can be used as an unbiased proxy to measure the relevance of a document to the given query. And to make a comprehensive comparison, we also performed the same evaluation on the normal click testing set in Figure 3 (b).



(a) P@1 ranking performance under different query frequency categories on the random bucket click set

(b) P@1 ranking performance under different query frequency categories on the normal click set

Figure 3: P@1 comparison between different click models over random bucket click set and normal click set.

As shown in Figure 3 (a) and (b), in the low query frequency category (query frequency <25), feature-based models outperformed the counting-based models on both random click set and normal click set. For those less frequent queries, counting-based models do not have enough observations to get a confident estimation of a document's relevance quality; while by leveraging the information across different observations via the same set of relevance-driven features, the feature-based models get a more accurate estimation of relevance for the documents in this category. With more observations available for a particular query, the relevance estimation quality of counting-based methods improves quickly and outperforms the simple feature-based methods on both testing sets. The reason for this slow improvement of simple feature-based methods is also due to the feature sharing: in terms of model complexity, counting-based models have more freedom to tune the parameters for each query-document pair; while feature-based models have to adjust the shared feature weights across all the training samples, such that it cannot arbitrarily fit all the observations. Our BSS model takes advantages of both counting-based and feature-based models by combining the perceived relevance, which is defined by the weighted sum of relevance-driven features as ${w^R}^\mathsf{T} f^R_{q,d}$, and the intrinsic relevance, which is modeled as query-document dependent parameters $w^R_{q,d}$, in a principled optimization framework. In BSS model, $w^R_{q,d}$ will be pushed close to zero for the less frequent queries, since there are no sufficient observations to get confident estimations for them, and therefore ${w^R}^\mathsf{T} f^R_{q,d}$ plays a more important role in estimating relevance. And when we get more observations for a particular query, $w^R_{q,d}$ is adjusted to further enhance the relevance estimation, which cannot be correctly predicted by the shared relevance features.

In Table 4 and Table 5, we list the ranking performance over all queries in the two testing sets, where a paired two-samples t-test is performed to validate the significance of improvement from the best performing method against the runner-up method under each performance metric. Since a large portion of testing queries in the random bucket click set belong to the less frequent category (only 29.3% queries appeared more than 100 times in the training set) comparing to the normal click set (76.7%), it becomes much more difficult for the purely counting-based methods to make accurate relevance estimation in the random bucket set. As we can observe in the results, although the counting-based UBM and DBN methods achieved better ranking performance than the simple feature-based models in the normal click set, their performance degraded on the random bucket set, due to the lack of observations. And as we have discussed earlier, by leveraging the feature-based and counting-based relevance estimations, BSS model outperformed all the other baseline methods on both data sets.

Table 4: Ranking performance on random bucket click set.

	LogisicReg	Examine	UBM	DBN	BSS
P@1	0.3696	0.3807	0.3603	0.3494	**0.4033***
P@2	0.3118	0.3348	0.3239	0.3194	**0.3509***
MAP	0.5628	0.6094	0.5951	0.5883	**0.6272***
MRR	0.5754	0.6154	0.6003	0.5939	**0.6330***

* indicates *p-value*<0.01

Table 5: Ranking performance on normal click set.

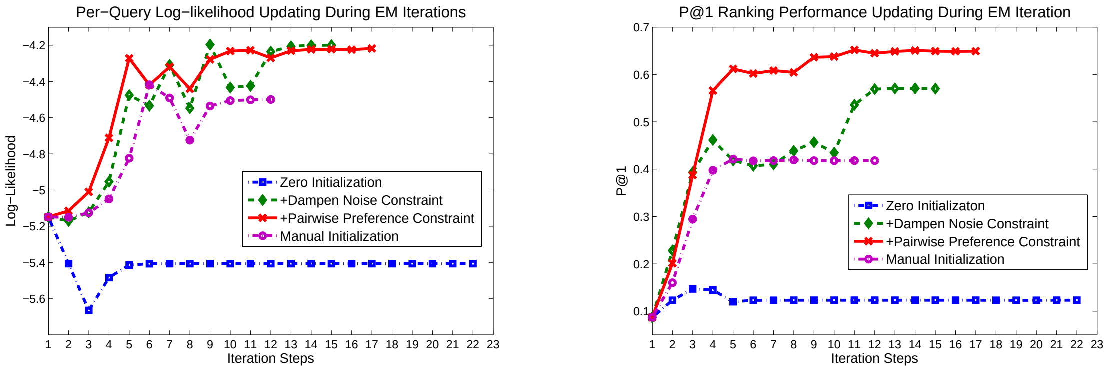
	LogisicReg	Examine	UBM	DBN	BSS
P@1	0.3623	0.3878	0.5316	0.4273	**0.5462**$^+$
P@2	0.3284	0.3058	0.3703	0.3166	**0.3823**$^+$
MAP	0.5981	0.5643	0.6688	0.5908	**0.6804**$^+$
MRR	0.6037	0.5786	0.6838	0.6047	**0.6964**$^+$

$^+$ indicates *p-value*<0.05

5.2.3 Estimated relevance as signals for learning-to-rank algorithm training

In this evaluation, we used the estimated relevance given by a click model as labels to extract ranking preference pairs of documents for training a learning-to-rank algorithm. We employed the pairwise RankSVM [13] as our basic learning-to-rank algorithm.

To stimulate the situation where we have to make prediction over some new documents before we can collect sufficient training clicks for each query, e.g., in news search, we only sampled 30% query log from each day in the normal training click set in this experiment. We estimated the click models on the new training set and generated the click preference pairs according to the predictions of each click model on this set. In particular, we ordered the documents under a given query according to their predicted relevance from a click model, and treated the top ranked document



(a) Per-query Log-likelihood updates

(b) P@1 ranking performance on training set updates

Figure 4: EM algorithm updating traces with different training settings.

Table 6: RankSVM performance on random bucket click set with different training signals.

	ori. click	UBM	DBN	BSS
P@1	0.3275	0.3823	0.3774	**0.3869**$^+$
P@2	0.3081	0.3411	0.3380	0.3411
MAP	0.5724	0.6130	0.6093	**0.6146**$^+$
MRR	0.5779	0.6187	0.6151	**0.6206**$^+$

$^+$ indicates *p-value*<0.05

Table 7: RankSVM performance on editorial judgments with different training signals.

	ori. click	UBM	DBN	BSS
P@1	0.5288	0.6346	0.6250	**0.6442**$^+$
P@2	0.4808	0.5313	0.5016	**0.5913**$^+$
MAP	0.6173	0.6944	0.6905	**0.7212**$^+$
MRR	0.6658	0.7546	0.7545	**0.7719**$^+$
NDCG@1	0.4570	0.5902	0.5774	**0.6016**$^+$
NDCG@5	0.5731	0.6830	0.6832	**0.7181**$^+$

$^+$ indicates *p-value*<0.05

as positive and others as negative. The preference pairs are extracted according to this predicted relevance labels under each query and fed into a RankSVM model. In addition, we also included a RankSVM trained on the preference pairs generated by the original clicks with the *skip-above* and *skip-next* click heuristics [14] in this training set as a baseline.

We compared the performance of RankSVM models trained by different relevance signals on both random bucket click set and editorial annotation set. In this experiment, we only included the UBM and DBN as the baseline click models since they performed much better than the simple feature-based Logistic Regression model and Examination Model on the normal click set according to Table 5.

As shown in Table 6 and Table 7, the training signals extracted from click models' output led to much better ranking performance of RankSVM than those extracted based on the simple click heuristics. In addition, though the RankSVM models trained on purely counting-based click models' output have comparable P@1 and NDCG@1 performance as that trained on BSS model's output, their predictions on the lower positions are much worse, e.g., lower MAP and NDCG@5. The main reason is that traditional click models only work in a pointwise way, and they cannot directly optimize the relative order of the predicted relevance; while in the proposed BSS model, we incorporated such ranking-oriented property via the *pairwise preference* constraint, which renders BSS model better capability of distinguishing the relative order among the candidate documents. As a result, the training signals extracted from BSS model's output are more informative for learning to rank algorithm training.

5.3 Effectiveness of Posterior Regularization

We now examine the effectiveness of a key component in the proposed BSS model, i.e., posterior regularization. As discussed in Section 4, there are two motivations for applying posterior regularization: one is to address the problem of "unidentifiability" in the proposed BSS model, and the other is to incorporate pairwise ranking preferences into click modeling. Below we validate the effectiveness of posterior regularization in achieving these two goals.

We first initialized all the model parameters, i.e., $\{w^R, w^C_{R=0}, w^C_{R=1}, w^E_{R=0}, w^E_{R=0}\}$, to be zero in our EM algorithm. We refer to this baseline as "zero initialization". And based on this initialization, we sequentially added the *noise dampening* constraint and *pairwise preference* constraint into the model to obtain two runs of model estimation using posterior regularized EM algorithm. An alternative way for solving the "unidentifiability" problem is to set priors over the model parameters such that we can guide the model to search in a desirable region. However, the difficulty of this approach for our BSS model is that it is unclear how to set proper priors on the model parameters for directly manipulating the probability $P(C|E, R)$, since this probability is defined over a set of different features via a logistic function. Thus to compare with such an approach, we manually set the initial value for the bias term in $w^C_{R=0}$ to be -1 and in $w^C_{R=1}$ to be 1, reflecting the assumption that most of the clicks should be explained by the relevance quality of a document rather than noise. We refer to this baseline as "manual initialization."

We plotted the per-query log-likelihood update trace and the corresponding P@1 ranking performance on the training set during EM iterations for these four different ways of learning our BSS model in Figure 4(a) and 4(b).

Effect of *dampen noise* constraint: as we can clearly observe from the EM update trace that without any specific parameter initialization or posterior regularization, EM failed to find a configuration which could improve the log-likelihood over the all-zero initialized model. Manual initialization helped the model identify better configurations. However, it is not a principled way for achieving so, and the scale of such hard-coded setting will directly bias the learned model. The proposed *dampen noise* constraint serves for the same purpose as manual initialization, but it gives model the freedom to learn such scale from data. As we can find from the updating trace, such constraint successfully led the model to a better configuration for both log-likelihood and P@1 ranking performance than the manual initialization.

Effect of *pairwise preference* constraint: with the *pairwise preference* constraint, which aims to enforce ranking-oriented requirement, the model's ranking capability is further improved, even though the log-likelihood did not gain too much. This is expected: log-likelihood defined in Eq (5) only considers the pointwise relevance estimation of each query-document pair, which does not count the relative order of relevance among the documents under the same query. The *pairwise preference* constraint explores such knowledge, which effectively improves ranking accuracy as shown in Figure 4(b). And we should note that such knowledge can hardly be encoded by manual initialization. Therefore, with the *pairwise preference* constraint, we solved the deficiency of traditional click models that the dependency relation among the clicked/skipped documents is discarded, and we are able to leverage the knowledge about pairwise preferences to further improve the relevance estimation accuracy.

5.4 Understanding User Behaviors with BSS

An interesting additional benefit of the proposed BSS model is that the learned feature weights reveal the influence of different factors on users' click behaviors, which is not available in existing click models. To explore this benefit, we list a subset of learned feature weights in Table 8.

Table 8: Feature weights learned by BSS model.

f^R	age	authority	title match	abs. match
w^R	-0.839	0.017	0.098	0.167
f^C	pos	dis. to last click	query length	bias
$w^C_{R=0}$	-1.133	-0.445	-3.659	-4.654
$w^C_{R=1}$	0.149	0.415	3.707	4.405
f^E	pos	# click	avg cont. sim.	bias
$w^E_{R=0}$	1.807	-0.418	2.947	5.325
$w^E_{R=1}$	-1.381	0.665	-2.237	3.266

The weights learned by our BSS model followed our intuition about their effects in influencing user's click decisions. For example, "age" is an important factor in news search: the most recent document (shorter age) is always preferred. Our BSS model correctly identified this negative correlation and put a relatively large weight over the age feature. The most interesting discovery by our model is the weights learned for the position feature in the click and examine events. In the click event, when the current document is irrelevant (i.e., $R = 0$), the weight for the position feature is quite negative, which indicates that a user is very likely to click on an irrelevant document when its displayed position is on the top, i.e., position-bias. But when the document is relevant (i.e., $R = 1$), the corresponding weight is closer to zero, which means user's click decision is not affected by the displayed positions of relevant documents. And in the examine event, the weight for the position feature is largely positive when the current document is irrelevant, which indicates users tend to further examine lower positions since they have not found satisfactory results. But when the current document is relevant, the weight becomes largely negative, which means users are inclined to stop further examination since their information need has been met by the relevant documents.

6. RELATED WORK

No previous work directly addressed the two deficiencies of existing click models, i.e., ignoring rich information conveyed in the document content when modeling clicks, and failing to exploit the relative order of relevance among the clicked/skipped documents. But there are several studies touched the problem of utilizing features in click modeling.

Richardson et al. [19] were the first to derive a content-based logistic regression model for predicting click throught rate by discounting the bias in lower positions via a position-specific multiplicative factor. However, they treated such discount factor as constant, which was only determined by positions, and thus it was independently estimated without considering the specific displayed documents and the related clicks. In [22], the authors also considered to introduce additional features to model click events; however, they used empirically tuned linear interpolation to combine the estimated relevance by a click model with external signals (e.g., BM25). Our method provides a more principled way for introducing rich descriptive features to formalize the dependency structure for both click and examine events within a query, and learn the optimal combination of those features from the data. Zhu et al. [23] realized the necessity of incorporating features into click models. However, they used the same set of general features (e.g., time and length of URL) to describe both click and examine events without distinguishing their specific effect in these two different events.

7. CONCLUSIONS

Click modeling is an important technique for exploiting search log data and is a crucial component in modern Web search engines. In this work, we proposed a general Bayesian Sequential State (BSS) model for addressing two deficiencies of existing click models, namely failing to utilize document content information for modeling clicks and not being optimized for distinguishing the relative order of relevance among the candidate documents. As our solution, a set of descriptive features and ranking-oriented pairwise preferences are encoded via a probabilistic graphical model, where the dependency relations among a document's relevance quality, examine and click events under a query are automatically captured from the data. Experiments on a large set of news search logs validate the effectiveness of the proposed BSS model comparing to several state-of-the-art click models, where content-based features help BSS model leverage information across different observations when the training set is limited, and pairwise preference constraint gives the model a more accurate estimate of relevance.

As we have shown in the experiment, the proposed BSS model provides an interesting way of understanding user's click behaviors by analyzing the learned weights on different features. With appropriate feature design, our model has the potential to help understand user behavior in various other aspects as well. As our future work, it would be meaningful to incorporate user-related features into our model, i.e., personalized BSS model, where different users will have their own weights over the designed features to reflect their unique search intents.

8. REFERENCES

[1] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10. ACM, 2006.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern information retrieval*, volume 463. ACM press New York, 1999.

[3] D. Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.

[4] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th international conference on World wide web*, pages 1–10. ACM, 2009.

[5] W. Chen, D. Wang, Y. Zhang, Z. Chen, A. Singla, and Q. Yang. A noise-aware click model for web search. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 313–322. ACM, 2012.

[6] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666, 2008.

[7] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the international conference on Web search and web data mining*, pages 87–94. ACM, 2008.

[8] G. E. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 331–338. ACM, 2008.

[9] J. Graça, K. Ganchev, and B. Taskar. Expectation maximization and posterior constraints. *Advances in Neural Information Processing Systems*, 20:569–576, 2007.

[10] L. A. Granka, T. Joachims, and G. Gay. Eye-tracking analysis of user behavior in www search. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 478–479. ACM, 2004.

[11] F. Guo, C. Liu, A. Kannan, T. Minka, M. Taylor, Y.-M. Wang, and C. Faloutsos. Click chain model in web search. In *Proceedings of the 18th international conference on World wide web*, pages 11–20. ACM, 2009.

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

[13] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

[14] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161. ACM, 2005.

[15] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. The MIT Press, 2009.

[16] L. Li, W. Chu, J. Langford, and X. Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 297–306. ACM, 2011.

[17] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[18] R. Neal and G. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. *Learning in Graphical Models*, 89:355–368, 1998.

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

[20] S. Shen, B. Hu, W. Chen, and Q. Yang. Personalized click model through collaborative filtering. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 323–332. ACM, 2012.

[21] Wikipedia. *Standard score*. `http://en.wikipedia.org/wiki/Standard_score`.

[22] Y. Zhang, D. Wang, G. Wang, W. Chen, Z. Zhang, B. Hu, and L. Zhang. Learning click models via probit bayesian inference. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 439–448. ACM, 2010.

[23] Z. A. Zhu, W. Chen, T. Minka, C. Zhu, and Z. Chen. A novel click model and its applications to online advertising. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 321–330. ACM, 2010.