Constructing Click Model for Mobile Search with Viewport Time

YUKUN ZHENG, JIAXIN MAO, YIQUN LIU, CHENG LUO, MIN ZHANG, and SHAOPING MA, Tsinghua University, China

A series of click models has been proposed to extract accurate and unbiased relevance feedback from valuable yet noisy click-through data in search logs. Previous works have shown that users search behavior in mobile and desktop scenarios are rather different in many aspects, therefore, the click models designed for desktop search may not be effective in the mobile context. To address this problem, we propose two novel click models for mobile search: (1) Mobile Click Model (MCM), which models click necessity bias and examination satisfaction bias; (2) Viewport Time Click Model (VTCM), which further extends MCM by utilizing the viewport time. Extensive experiments on large-scale real mobile search logs show that: (1) MCM and VTCM outperform existing models in predicting users' clicks and estimating result relevance; (2) MCM and VTCM can extract richer information, such as the click necessity of search results and the probability of user satisfaction, from mobile click logs; (3) By modeling the viewport time distributions of heterogeneous results, VTCM can bring a significant improvement over MCM in click prediction and relevance estimation tasks. Our proposed click models can help better understand user behavior patterns in mobile search and improve the ranking performance of mobile search engines.

CCS Concepts: • **Information systems** → **Web search engines**; *Users and interactive retrieval*; *Retrieval on mobile devices*;

Additional Key Words and Phrases: Click model, viewport time, mobile search, web search

ACM Reference format:
Yukun Zheng, Jiaxin Mao, Yiqun Liu, Cheng Luo, Min Zhang, and Shaoping Ma. 2019. Constructing Click Model for Mobile Search with Viewport Time. *ACM Trans. Inf. Syst.* 37, 4, Article 43 (September 2019), 34 pages.
https://doi.org/10.1145/3360486

1 INTRODUCTION

Previous studies showed that user clicks can be used as implicit relevance feedback to improve the ranking of search results [17]. However, clicks on a result are inherently stochastic and

This work is supported by the National Key Research and Development Program of China (2018YFC0831700) and Natural Science Foundation of China (Grant No. 61622208, 61732008, 61532011). This work is also part of NExT++ project, supported by the National Research Foundation, Prime Minister's Office, Singapore under its IRC@Singapore Funding Initiative.
This article is an extension of Mao et al. [30]. Compared with the previous conference version, it introduces a new Viewport Time Click Model (VTCM) that incorporates viewport time information. It also includes an extensive experimental assessment of the new model and compares the performance with a number of existing models including MCM.
Authors' addresses: Y. Zheng, J. Mao, Y. Liu (corresponding author), C. Luo, M. Zhang, and S. Ma, Tsinghua University, Beijing, 100084, China; emails: {zhengyk13, maojiaxin}@gmail.com, {yiqunliu, chengluo, z-m, msp}@tsinghua.edu.cn.

1046-8188/2019/09-ART43 $15.00
https://doi.org/10.1145/3360486

systematically biased by factors such as the position [8, 17] and presentation style [4, 38] of the result. Therefore, a number of click models (see Reference [6] for an overview) have been proposed to model user click behavior as a stochastic process and obtain unbiased relevance feedback from the biased click logs.

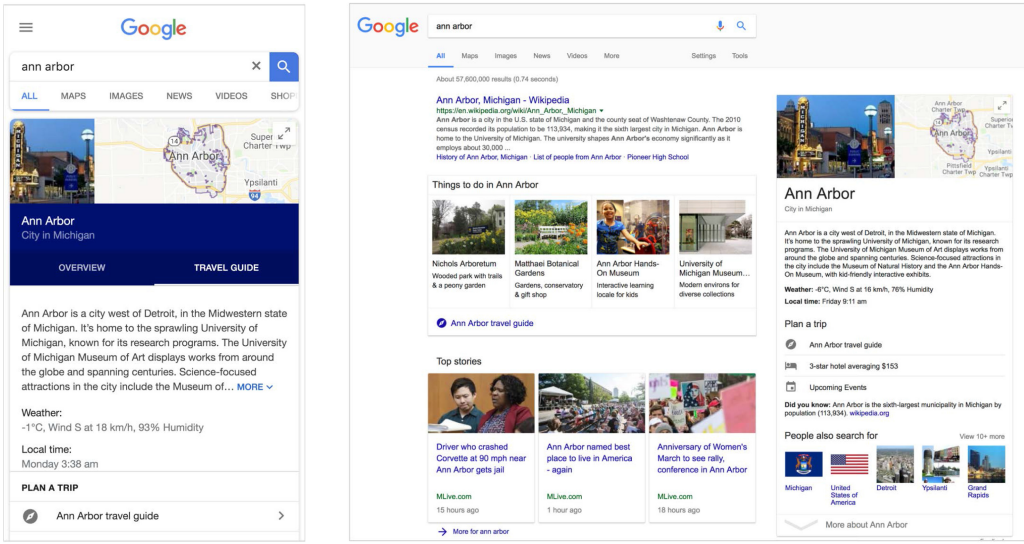
The performance of a click model depends heavily on making correct assumptions on user search behavior. By assuming a user will examine and click the results on the search engine result page (SERP) in a certain way, a click model can estimate how different kinds of biases affect users' click actions and derive unbiased relevance feedback from click logs. However, user search behavior in the mobile environment are different from those in the desktop context. For example, previous studies suggest that users will pay more attention to the top-ranked results and scan fewer results on a small screen [20]; relevance judgments for documents are also affected by search devices [36]. Therefore, the existing click models originally designed for the desktop environment may not be as effective in the mobile search context. We need to refine the existing behavioral assumptions of click models to adapt to the shift from desktop to mobile.

One of the factors that may alter user behavior in the mobile environment is the heterogeneity of search results. Today's search engines return richer results than the homogeneous 10 blue links on both mobile and desktop. The heterogeneous results have a larger impact on user interaction behavior on mobile SERPs because: (1) Compared to desktop search, direct answer and knowledge card results are federated into mobile SERPs more frequently. In many circumstances, these results present useful information on the SERP and users do not need to click the hyperlinks to visit the corresponding landing pages. While loading a page on mobile devices may take a longer time than on desktop devices, this strategy helps to reduce users' interaction costs as well as data usage on mobile; (2) Due to the limit of screen size, the heterogeneous results are usually injected into the main ranking list and often occupy a large proportion of user viewport.[1] In a recent study, Luo et al. [28] showed these two factors may affect user behavior in mobile search and proposed to incorporate them in the evaluation of mobile search engines.

As an example, we show two SERPs for the same query, *ann arbor*, on mobile and desktop in Figure 1. Compared with the desktop SERP in Figure 1(b) that displays the knowledge graph result on the right side, the knowledge graph result is placed at the first position in the mobile SERP (Figure 1(a)) and occupies almost the whole initial viewport. This result is highly likely to be examined by users and affect their following actions. The knowledge graph result contains a brief introduction to the city, as well as information about the weather and local time. A user who wants to gather some basic information about Ann Arbor will find the knowledge graph result relevant and useful even without clicking it. She may even feel satisfied and leave the SERP just after examining the first knowledge graph result. In this case, an existing click model will: (1) mistakenly regard the skipping (i.e., no click) behavior on the first result as a negative relevance feedback; (2) ignore the cut-off effect [26]—that the user can be satisfied with the non-clicked knowledge graph result but still assume the user will scan the following results.

To address these problems in the mobile search context, we propose a novel click model named Mobile Click Model (MCM). The proposed MCM assumes that: (1) Some types of search results (e.g., the knowledge graph and direct answer results) have lower click necessity than others, which means that they can fulfill users' information needs without requiring any clicks (click necessity bias); (2) A user can be satisfied after *examining* a search result with low click necessity, because this kind of result is designed to satisfy users' common information needs directly on SERPs (examination satisfaction bias).

[1]The portion of the SERP that is visible on the screen of the mobile phone at a certain time.



(a) Mobile SERP (b) Desktop SERP

Fig. 1. Examples of SERPs on (a) mobile and (b) desktop from Google. Only the content in the initial viewport is shown.

Besides user clicks, we can also leverage the viewport time signal in mobile search. Several works [22, 23] looked into the viewport time in mobile search and found that it has a strong correlation with user attention and can be adopted as a kind of effective feature to predict user satisfaction. Viewport coordinates can be recorded by the search engine when specific user behaviors happen, such as entering or leaving the SERP, scrolling up and down, and so on. Thus, we can unobtrusively collect viewport coordinates and calculate the viewport time of search results[2] on the SERP. Therefore, we further extend MCM by incorporating the viewport time of search results and propose a new click model named Viewport Time Click Model (VTCM). In VTCM, we treat viewport time as the second observed variable besides the user click. We will introduce how we incorporate click necessity bias and examination satisfaction bias into MCM as well as how VTCM models the viewport time distributions in Section 3.

Through extensive experiments on a large-scale mobile search log from a popular commercial search engine in China, we first show that MCM and VTCM can effectively infer the parameters for click necessity and examination satisfaction, along with the parameters for relevance and click satisfaction, from users' interaction logs with heterogeneous mobile SERPs. With the viewport coordinate data available in the mobile search log, we further show that VTCM can effectively estimate the viewport time distributions under different conditions of examination behavior, user clicks, and examination satisfaction. With these parameters learned from logs, we can: (1) improve the ranking of heterogeneous results in mobile search; (2) analyze how users interact with a certain type of vertical results. The experiment results also show that MCM and VTCM achieve better performance in both click prediction and relevance estimation tasks than the baseline click models that are not specifically designed for the mobile environment.

The rest of the article is organized as follows: We first provide an overview of the background of mobile search and click models in Section 2. In Section 3, we will formally introduce MCM and

[2]The duration of search results exposed in the viewport.

VTCM and then compare them with existing click models. The experiment setup and results are presented in Section 4. Finally, we conclude the article and discuss directions for future work in Section 5.

2 RELATED WORK AND BACKGROUND

2.1 Search Behavior on Mobile

With the rise of mobile search, understanding user search behavior on mobile devices becomes increasingly important. Existing research has characterized the differences between desktop search and mobile search in various aspects.

First, compared to desktop search, mobile search is often conducted to fulfill different types of information needs, in diverse contexts. Yi et al. [43] and Kamvar et al. [19] are among the first who spotted a difference in the distribution of query categories across difference search devices. Song et al. [35] further found that the information needs of mobile searchers varied at different times of the day. They also showed that a mobile user tended to search at different locations and users' click preferences changed with the search devices. Recently, Harvey and Pointon [14] suggested that users often use mobile devices to search in an "on the go" context, where they might be interrupted or distracted. They conducted a user study to assess the impact of these "fragmented attention" situations on user search behavior and performance. The differences in search contexts and information needs on mobile and desktop suggest that the mobile search engine should return different results to satisfy mobile searchers. Therefore, it is crucial to develop new methods to extract relevance feedback from mobile search logs.

Second, the user interface (UI) of mobile search is very different from that of desktop search. Unlike a desktop PC with a large display (13 to 30 inches) as well as a mouse and a keyboard as input devices, a mobile phone usually has a much smaller screen (4 to 5 inches) and responds to a variety of touch interactions, including swiping, zooming, and on-screen text input. Previous works studied how the differences in UIs affect user search behavior on mobile and desktop. Regarding the differences in input interactions, Kamvar and Baluja [18] and Song et al. [35] showed that while the query length was not significantly different on mobile and desktop, the mobile searcher tended to issue fewer queries in a session than the desktop searcher; Guo et al. [13] proposed to use the mobile touch interactions as features to estimate the relevance of mobile search results and identified some similarities and differences between users' fine-grain interactions on the landing pages in both desktop and mobile environments. However, the difference in screen size may impose more efforts for the mobile searchers to gather the same amount of information. Kim et al. [20] conducted an eye-tracking study to compare users' SERP scanning patterns on small screens and large screens. They found that on small screens, users gave more attention to top-ranked results and exhibited a more linear scanning pattern. Recently, Ong et al. [32] found that users used different search strategies to adapt to the SERPs with varying Information Scent Levels and Information Scent Patterns [42] on mobile and desktop. These studies showed that user search behavior on mobile devices was different from that in traditional desktop settings, therefore the click models that were originally designed to model user click behavior in desktop search need to be adapted for mobile environment.

Third, today's mobile search engines will return more diverse results to cope with some specific information needs (e.g., checking the weather forecast or looking for a restaurant nearby) and reduce users' interaction cost in mobile environment. These heterogeneous *vertical* results may alter user search behavior on mobile. For desktop search, Liu et al. [26] conducted a dedicated eye-tracking study to analyze the effects of different types of vertical results on users' examination and click behavior on SERPs. For mobile search, Lagun et al. [22] studied how knowledge

graph results affected users' attention and satisfaction. Their results showed that when a relevant knowledge graph result was presented, the user would pay less attention to the results below it, spend less time on the whole SERP, and feel more satisfied in the search. They also used an eye-tracker to measure user gaze time on each search result and found that users paid more attention to the second and third results than the first results in mobile search, which is different from the findings in the eye-tracking studies conducted in desktop search settings (e.g., References [11, 17]). Williams et al. [40] found that in mobile search, the direct answer results often led to *good abandonment*, where the user was directly satisfied by the SERP without clicking any hyperlinks, and they proposed a gesture model that utilizes viewport time features to predict user satisfaction for the abandoned queries. Williams et al. [41] further looked into how different types of answer verticals affect user behavior in mobile search. These findings emphasized the importance of modeling the heterogeneity of search results in building click models for mobile search.

Fourth, the different context between desktop and mobile devices leads to the different types of available user behavior data collected in the search logs. On desktop devices, the hover with mouse cursor can be collected by the search engine system and serve as the user attention in the search process, which is shown to be useful in several research tasks, such as inferring result relevance [31] and predicting user clicks [15]. However, there is no mouse cursor in the mobile environment, so several works [22, 23] suggested using viewport time as an alternative to serving as a kind of user attention. As the screens of mobile devices are much smaller than desktop devices and can usually contain only one to three search results at the same time, Lagun et al. [22] looked into the viewport time in mobile search and showed its strong correlation with users' eye gaze. Lagun et al. [23] found that the viewport data is useful for measuring user attention on both information-rich advertisement and organic search results in the mobile environment. Mao et al. [29] study the relationship between the user's usefulness feedback and their search behavior, showing that the viewport time features can be used to estimate usefulness when user clicks are absent. Based on large-scale mobile search logs, Wang et al. [39] looked into the relationship between the viewport time and user behavior, such as user examination and clicks, showing that the viewport time that users spend on viewing clicked results has different distribution from the one spent on viewing results without any user click. In a word, although the viewport time of mobile search results is inherently stochastic, these works showed that it is systematically related to types of results, examination behavior, user clicks, and satisfaction. Therefore, in this work, we incorporate the viewport time as an additional signal in click models.

2.2 Click Models for Web Search

In this section, we will first present some definitions and notations used in this article and introduce some existing click models, along with their corresponding behavioral assumptions, in these notations. We will also introduce existing research on click models that has considered the heterogeneity of search results and richer user behavior information.

When a user submits a *query* q to the search engine in a *session* s, a SERP that consists of M ranked *search results*, $(d_1, d_2, \ldots, d_M)$, will be returned to the user. Usually, M is set to 10, because there are usually 10 results on the first page. d_i denotes the search results ranked at position i. d_i can be an *organic* result or one of different types of *vertical* results. We use v_i to denote the *type* of d_i. M binary random variables $(C_1, C_2, \ldots, C_M)$ are used to indicate whether the user *click* d_i ($C_i = 1$) or *skip* d_i ($C_i = 0$). C_i can be observed in the search log. A click model is usually a probabilistic generative model of the click sequence $(C_1, C_2, \ldots, C_M)$ that models the joint distribution $P(C_1, C_2, \ldots, C_M)$.

Originally, click models were proposed to explain the position bias that users are more likely to click top-ranked results, because these results are more likely to be *examined*. To model this bias

caused by differences in examination likelihood at different ranks, the *Examination Hypothesis* was formulated by Richardson et al. [33] in predicting the click-through rate of ads and Craswell et al. [8] in modeling the position bias in web search. This hypothesis assumes that a user will click a search result if and only if she examined the result and was attracted by it:

$$C_i = 1 \iff E_i = 1 \wedge A_i = 1. \tag{1}$$

E_i and A_i are binary random variables. Unlike C_i, they are *latent* variables that cannot be observed directly from search logs. $E_i = 1$ indicates the user examined d_i and otherwise $E_i = 0$. $A_i = 1$ means the search result can attract the user's click whenever she examines it. A_i is usually considered as fully determined by the relevance between query q and result d_i:

$$P(A_i = 1) = \alpha_{q,d_i}. \tag{2}$$

Therefore, A_i is independent of E_i, and the click probability of d_i can be computed as:

$$P(C_i = 1) = P(E_i = 1) \cdot P(A_i = 1). \tag{3}$$

A series of click models have different implementations of $P(E_i)$. For example, the *cascade model* proposed by Craswell et al. [8] assumes a user will examine the search results sequentially from top to bottom until she clicks a result. Therefore, $P(E_i = 1) = 1, \forall i \leq j$, where j is the rank of last clicked results in the session. Guo et al. [12] extended the cascade model to multi-click sessions by assuming that the user will continue to examine next results after clicking a result at position i with a probability of λ_i. Dupret and Piwowarski [9] proposed the User Browsing Model (UBM), which assumes that $P(E_i)$ depends on the current position i and its distance d to a previously clicked result:

$$P(E_i = 1) = \gamma_{i,d}. \tag{4}$$

Chapelle and Zhang [3] used additional binary variables S_i to denote the user's *satisfaction* after clicking a result. If d_i is clicked ($C_i = 1$), then S_i only depends on the query q and result d_i and is considered as an additional signal for relevance:

$$P(S_i = 1|C_i = 0) = 0, \tag{5}$$

$$P(S_i = 1|C_i = 1) = s_{q,d_i}. \tag{6}$$

They also assumed that a user will scan the SERP linearly but they allowed the user to leave the SERP, not examining lower-ranked search results, when she is satisfied by a result d_i ($S_i = 1$) or choose to abandon the query with a probability $1 - \gamma$:

$$P(E_1 = 1) = 1, \tag{7}$$

$$P(E_i = 1|S_{i-1} = 1) = 0, \tag{8}$$

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

$$P(E_i = 1|S_{i-1} = 0, E_{i-1} = 1) = \gamma. \tag{10}$$

With the development of deep learning, Borisov et al. [1] proposed Neural Click Model (NCM) to model the sequence of user actions in the search process and achieved better performance in the click prediction task than several popular click models with the *probabilistic graphical model* (PGM) framework including UBM [9] and DBN [3].

With the emergence of vertical results and federated search, some existing efforts in desktop web search tried to incorporate the influence of different vertical results into click models. Chen et al. [4] considered the *attention bias* that if d_i is a vertical result, it may have a higher examination probability $P(E_i = 1)$ and the *exploration bias* that the user may choose not to examine any organic results if she clicked a vertical with a certain probability $e(s)$ in the session s. Chuklin et al. [7]

addressed this problem by assuming that a session is associated with a pre-defined intent $I(s)$. This intent and the type of result v_i will affect the examination probability $P(E_i)$ and click attractiveness $P(A_i)$. One can incorporate the influence of intents and result types into a click model that follows the examination hypothesis (Equation (1)). For example, the UBM can be enhanced in the following way:

$$P(E_i = 1) = \gamma_{i,d}(I(s), v_i), \tag{11}$$

$$P(A_i = 1) = \alpha_{q,d_i}(I(s)). \tag{12}$$

The Vertical-aware Click Model (VCM) proposed by Wang et al. [38] further modeled how the presence of different types of verticals affects both the examination probabilities and examination order of the results on SERP. However, VCM can only model the influence of the *first* vertical result on the SERP, making it not suitable for the mobile search scenario where a SERP usually contains multiple vertical results.

These studies all focused on modeling how vertical results affect the examination probability $P(E_i)$, but none of them addressed the problem that some types of vertical results can satisfy users without requiring any clicks. Chuklin and de Rijke [5] proposed the Clicks, Attention and Satisfaction (CAS) model to address the good abandonments problem in the desktop search environment by assuming that satisfaction is cumulative and happens during the query session. Such skips on good results with low click necessity are rather common in mobile search, which motivates us to propose a new click model to cope with the corresponding click necessity bias and examination satisfaction bias in mobile environment. Different from the CAS model, our mobile click models estimate the click necessity of a search result as one of its attribute variables as same as attractiveness and satisfaction.

Another line of research tries to incorporate richer user behavior information into click models. Wang et al. [37] first incorporated non-sequential behavior into click models and proposed Partially Sequential Click Model (PSCM), while Liu et al. [25] extended PSCM by capturing the temporal information of user behavior and proposed Time-Aware Click Model (TACM), which models the relationship between click dwell time and user satisfaction. Liu et al. [27] proposed to enhance the estimation of examination by incorporating mouse movement information into existing click models. In this work, as the viewport time information has a strong correlation with user examination and satisfaction and is available in the mobile search environment, we incorporate it into the click model to promote the performance in both predicting user clicks and estimating result relevance.

3 MOBILE CLICK MODEL

3.1 Modeling Biases

We first formally introduce the click necessity bias and examination satisfaction bias as well as how we incorporate them into click models.

- **Click Necessity Bias**: *Some types of search results (e.g., the knowledge graph and direct answer results) have low click necessity, because they can satisfy users' information needs without requiring any clicks, which will lower the click probabilities of these results.*

To model the click necessity bias, we introduce a binary variable N_i for the click necessity of each result d_i. $N_i = 1$ indicates that a user *must* click the result to get the useful information in it, and $N_i = 0$ indicates that a user can be satisfied directly by reading or interacting with the snippet on the SERP. For example, the knowledge graph result that presents rich information in Figure 1(a) tends to be of lower click necessity than an organic result that contains less useful information.

We extend the examination hypothesis (Equation (1)) as:

$$C_i = 1 \iff E_i = 1 \wedge A_i = 1 \wedge N_i = 1. \tag{13}$$

A user will click a search result if and only if: (1) she examined it; (2) it is attractive; and (3) she needs to click it to get useful information. We further assume that N_i only depends on the type of search results v_i:

$$P(N_i = 1) = \beta_{v_i}. \tag{14}$$

We acknowledge that $P(N_i = 1)$ may also be affected by other factors such as user intent and relevance between the query and result, but we choose to use this simplified assumption and leave the exploration of how to model $P(N_i = 1)$ for future work.

By incorporating the click necessity bias, we can avoid the negative feedback caused by the good skips on the results with low click necessity. However, we also need to define positive signals, other than clicks, for these results. Therefore, we propose the examination satisfaction bias:

- **Examination Satisfaction Bias**: *A user can feel satisfied and leave the SERP after examining a search result that is both attractive and with low click necessity.*

We use a binary variable S_i^E to denote whether the user is satisfied just by *examining* result d_i (examination satisfaction), S_i^C to denote whether the user is satisfied after *clicking* it (click satisfaction). We further use S_i to denote a user's *state of satisfaction* after position i. We assume that: (1) a user will stay satisfied once she encountered either an examination satisfaction event ($S_i^E = 1$) or a click satisfaction event ($S_i^C = 1$); (2) if the user is in the satisfied state $S_i = 1$, she will not examine follow-up results. Therefore, we have:

$$S_i = 1 \iff S_{i-1} = 1 \vee (S_i^E = 1 \vee S_i^C = 1), \tag{15}$$

$$P(E_i = 1 | S_{i-1} = 1) = 0. \tag{16}$$

Because $S_i = 1 \Rightarrow S_{i+1} = 1$, we have $\forall j > i, P(E_j = 1 | S_i = 1) = 0$. By adding the satisfaction state variable S_i, we allow the click/examination satisfaction event at position i (S_i^C and S_i^E) to influence all the follow-up examination events (E_j, where $j > i$).

The click satisfaction ($S_i^C = 1$) can happen when a result is clicked, while the examination satisfaction ($S_i^E = 1$) can only occur when a result is examined ($E_i = 1$), attracts user's attention ($A_i = 1$), and it does not need to be clicked ($N_i = 0$). Similar to DBN, we assume that S_i^E and S_i^C are governed by the parameters that associate with the relevance between q and d_i:

$$P(S_i^C = 1 | C_i = 1) = s_{q,d_i}^C, \tag{17}$$

$$P(S_i^E = 1 | E_i = 1, A_i = 1, N_i = 0) = s_{q,d_i}^E. \tag{18}$$

By incorporating the examination satisfaction bias, we can give credits to the search results that have low click necessity but can provide relevant and useful information for users when there are no clicks below it. We hope that capturing these signals can help us rank the results with low click necessity more properly in mobile search.

3.2 Modeling Viewport Time

We treat viewport time V_i as an observed and continuous variable in VTCM model and adopt continuous probability distributions to model the distributions of results' viewport time under different conditions of examination behavior, user clicks, and examination satisfaction. Besides, we assume that the viewport time probabilities are also related to the type v_i of the search result for which the user usually pays different attention to different types of search results [23, 38]. Based on the hypotheses of MCM, there are four possible conditions for the ith result in the SERP: (1) The

user did not examine it ($E_i = 0$); (2) The user examined it and continued examining another result in the SERP ($E_i = 1, C_i = 0, S_i^E = 0$); (3) The user examined it, felt satisfied, and left ($E_i = 1, C_i = 0, S_i^E = 1$); (4) The user examined and clicked it ($E_i = 1, C_i = 1, S_i^E = 0$). Therefore, we propose two versions of VTCM to model the distributions of viewport time. In the first version of VTCM named VTCM$_e$, we classify the four conditions into two types: examined (Condition 1) and not examined (Conditions 2, 3, 4). Therefore, for the viewport time t_i of the ith search result in the SERP, we have two conditional probabilities with respect to the two types of conditions, i.e., Equations (19) and (20).

$$P(V_i = t_i | E_i = 0) = f_{v_i}^{E=0}(t_i), \tag{19}$$

$$P(V_i = t_i | E_i = 1) = f_{v_i}^{E=1}(t_i). \tag{20}$$

In the second version of VTCM named VTCM$_c$, we take examination behavior, user clicks, and examination satisfaction into account to model the viewport time distributions and adopt four independent conditional probabilities for those conditions, which are listed as follows:

$$P(V_i = t_i | E_i = 0) = f_{v_i}^{E=0}(t_i), \tag{21}$$

$$P(V_i = t_i | E_i = 1, C_i = 0, S_i^E = 0) = f_{v_i}^{E=1, C=0, S^E=0}(t_i), \tag{22}$$

$$P(V_i = t_i | E_i = 1, C_i = 1, S_i^E = 0) = f_{v_i}^{E=1, C=1, S^E=0}(t_i), \tag{23}$$

$$P(V_i = t_i | E_i = 1, C_i = 0, S_i^E = 1) = f_{v_i}^{E=1, C=0, S^E=1}(t_i). \tag{24}$$

For the viewport time distribution function f in both VTCM$_e$ and VTCM$_c$, we try three continuous probability distribution functions: *log-normal*, *gamma* and *Weibull*, whose probability density functions are listed here:

$$f_{log-normal}(t; \mu, \sigma) = \frac{1}{t\sigma\sqrt{2\pi}} e^{-\frac{(\ln t - \mu)^2}{2\sigma^2}}, \tag{25}$$

$$f_{gamma}(t; k, \theta) = \frac{1}{\Gamma(k)\theta^k} t^{k-1} e^{-t/\theta}, \tag{26}$$

$$f_{Weibull}(t; \lambda, k) = \frac{k}{\lambda}\left(\frac{t}{\lambda}\right)^{k-1} e^{-(t/\lambda)^k}. \tag{27}$$

By incorporating the viewport time information, we can correct the estimated examination probabilities of search results in a specific search process and extend the assumptions of the click model by establishing the relationship among viewport time, examination satisfaction, and user click behavior. We hope that it can help model user behavior and estimate latent variables in the model, e.g., the relevance and examination satisfaction of search results, more effectively. More analysis of viewport time distributions learned by VTCM will be discussed in Section 4.3.

3.3 Mobile Click Model

Besides incorporating the click necessity bias and examination satisfaction bias, we can use different functions for $P(E_i)$ and $P(A_i)$, which will be equivalent to incorporating the click necessity bias and examination bias into different click models. In this work, we use UBM's implementation of $P(A_i)$ and $P(E_i)$ (Equations (2) and (4)) because: (1) it performs well in the click prediction task; (2) the computation of $P(E_i = 1)$ is fully determined by observable variables $C_j, j < i$, which simplifies the inference of posterior.

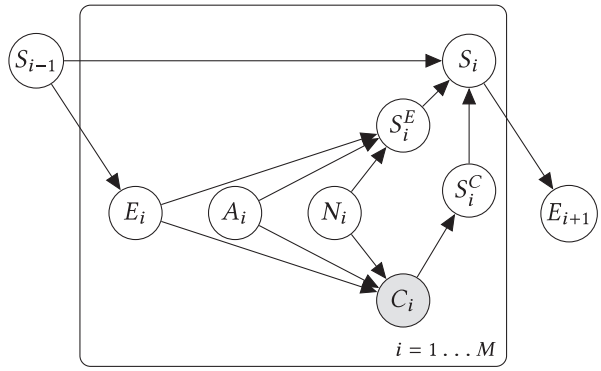


Fig. 2. The Bayesian network structure of Mobile Click Model (MCM). C_i is the only observed variable.

We call the derived model Mobile Click Model (MCM) and illustrate it in Figure 2. In this model, only C_i can be observed in logs and only S_i will influence users' further behavior. The conditional probabilities of C_i and the latent variables $\{E_i, A_i, N_i, S_i^E, S_i^C, S_i\}$ are given as follows:

$$P(E_i = 1|S_{i-1} = 1) = 0, \tag{28}$$
$$P(E_i = 1|S_{i-1} = 0) = \gamma_{i,d}, \tag{29}$$
$$P(A_i = 1) = \alpha_{q,d_i}, \tag{30}$$
$$P(N_i = 1) = \beta_{v_i}, \tag{31}$$
$$C_i = 1 \iff E_i = 1 \wedge A_i = 1 \wedge N_i = 1, \tag{32}$$
$$P(S_i^E = 1|\neg(E_i = 1 \wedge A_i = 1 \wedge N_i = 0)) = 0, \tag{33}$$
$$P(S_i^E = 1|E_i = 1 \wedge A_i = 1 \wedge N_i = 0) = s^E_{q,d_i}, \tag{34}$$
$$P(S_i^C = 1|C_i = 0) = 0, \tag{35}$$
$$P(S_i^C = 1|C_i = 1) = s^C_{q,d_i}, \tag{36}$$
$$S_i = 1 \iff S_{i-1} = 1 \vee (S_i^E = 1 \vee S_i^C = 1). \tag{37}$$

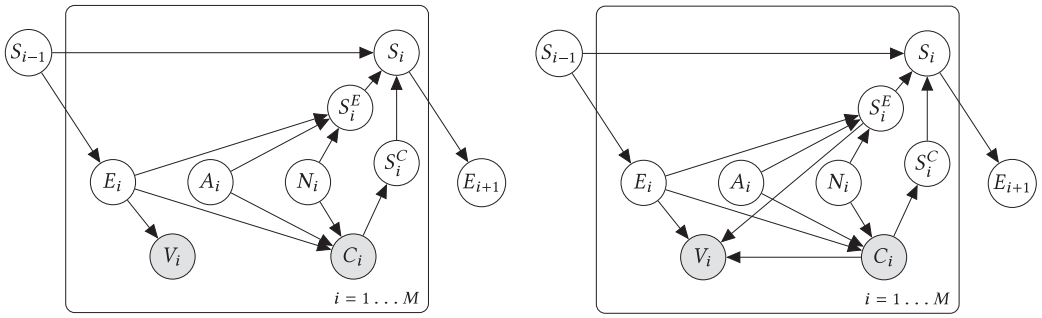
3.4 Viewport Time Click Model

We introduce viewport time information into VTCM and illustrate it in Figure 3, where C_i and V_i are two observed variables. In VTCM$_e$, the conditional probabilities of V_i are only related to E_i (Equations (19) and (20)), while the conditional probabilities of V_i are related to E_i, C_i, and S_i^E in VTCM$_c$ (Equations (21)–(24)). In both VTCM*e* and VTCM*c*, the probabilities of C_i and latent variables $\{E_i, A_i, N_i, S_i^E, S_i^C, S_i\}$ are given as same as MCM (Equations (28)–(37)).

3.5 Parameter Update

The parameters of MCM are $\{\alpha, \beta, \gamma, s^E, s^C\}$, while VTCM includes additional parameters Θ for the viewport time distributions. The maximum likelihood estimates of these parameters can be learned from click logs by using the Expectation-Maximization (EM) algorithm. The objective of VTCM changes, because V_i is introduced as the second observed variable. The objectives of MCM and VTCM are to maximize the probabilities given as follows:

$$\mathbb{J}_{MCM} = \prod_{s \in S} P(C_1^s, \ldots, C_M^s|\alpha, \beta, \gamma, s^E, s^C), \tag{38}$$



(a) VTCM$_e$ (b) VTCM$_c$

Fig. 3. The Bayesian network structure of Viewport Time Click Model (VTCM). C_i and V_i are two observed variables. In VTCM$_e$, the conditional probability of V_i relies on the prior probability of E_i, while in VTCM$_c$, the conditional probability of V_i relies on the joint prior probability of E_i, C_i, and S_i^E.

$$\mathbb{J}_{VTCM} = \prod_{s \in S} P(C_1^s, \ldots, C_M^s, V_1^s, \ldots, V_M^s | \alpha, \beta, \gamma, s^E, s^C, \Theta), \tag{39}$$

where S is the set of query sessions in the training data and M is the number of documents in a query session. Please refer to the Appendix for a detailed derivation of the E-step and M-step.

After learning the parameters, we can use them to compute a relevance score for each mobile search result in the logs. This score can be used to rank the search results according to users' implicit relevance feedback. For MCM and VTCM, we use Equation (40) to compute the predicted relevance score, where we treat the probability that the user feels satisfied to the result after examining it as the relevance score of (q, d_i):

$$\begin{aligned} relevance(q, d_i) &\triangleq P(S_i = 1 | E_i = 1) \\ &= \alpha_{q,d_i} \left[\beta_{v_i} s^C_{q,d_i} + (1 - \beta_{v_i}) s^E_{q,d_i} \right]. \end{aligned} \tag{40}$$

3.6 Comparisons with Existing Click Models

We compare the behavioral assumptions of MCM and VTCM with the ones of some existing click models in Table 1.

First, we compare MCM with two widely used click models: DBN and UBM. Compared to UBM, MCM takes the click and examination satisfaction into consideration. Therefore, in MCM, the examination probability $P(E_i = 1)$ is not only dependent on user click behavior on previous results $(C_1, C_2, \ldots, C_{i-1})$ but also influenced by the relevance of these results captured by the satisfaction parameters s^C_{q,d_j} and s^E_{q,d_j}, for all $j < i$. Compared to DBN, MCM relaxes the strict *cascade hypothesis* in examination that $E_{i-1} = 0 \Rightarrow E_i = 0$. Instead, MCM allows skips in an examination sequence as the UBM does. From this perspective, MCM can be regarded as an effort to unify these two classic click models.

We also compare MCM with previous efforts on incorporating the heterogeneity of search results into click models [4, 7, 38]. The existing studies in desktop search mainly focused on modeling the influence of heterogeneous results on user examination behavior (attention bias) and the preference to a certain type of vertical results caused by different search intents (search intent bias). None of them addressed the click necessity bias and examination satisfaction bias that are more common in mobile search.

Compared to MCM and other existing click models, VTCM further utilizes the viewport time information. During the inference process, the posterior examination probability $P(E_i = 1)$ and

Table 1. Comparisons between MCM, VTCM, and Some Existing Click Models

	DBN [3]	UBM [9]	Chen et al. [4]	Chuklin et al. [7]	Wang et al. [38]	MCM	VTCM
allow skip examination		✓	✓	✓	✓	✓	✓
click satisfaction	✓		✓[3]			✓	✓
attention bias			✓	✓	✓		
search intent bias				✓			
click necessity bias						✓	✓
examination satisfaction bias						✓	✓
viewport time							✓

satisfaction probability $P(S_i = 1)$ will be affected by the viewport time information, which can help VTCM estimate parameters more effectively and achieve better click prediction performance.

4 EXPERIMENTS

We conduct a series of experiments on large-scale search logs collected from a popular Chinese mobile search engine to answer the following research questions:

- **RQ1:** Can VTCM model the viewport time distributions of heterogeneous mobile search results effectively?
- **RQ2:** Do MCM and VTCM have better click prediction ability in the mobile environment than the baseline models?
- **RQ3:** Can MCM and VTCM provide better relevance estimations of mobile search results than the baseline models?
- **RQ4:** How do MCM and VTCM model the click necessity and examination satisfaction probability of heterogeneous mobile search results?

4.1 Experimental Setup

4.1.1 Datasets. The search logs used in this study were sampled from real mobile search logs of Sogou.com, a popular Chinese search engine. The search log for a session s consists of a query q, 10 URLs of search results, a 10-dimensional binary click vector $(C_1, C_2, \ldots, C_{10})$, a 10-dimensional viewport time vector $(V_1, V_2, \ldots, V_{10})$, and 10 `vertical_ids` for the search results. For the viewport time vector, we inject Javascript into SERPs to log users' scrolling and tab-switch actions. (See Section 4.1.2 for details about how we compute the viewport time of search results.) In this study, we use the corresponding `vertical_id` to indicate the type (v_i) of a search result (d_i). We note that this is a fine-grain categorization of search results, because there are thousands of unique `vertical_ids` in the logs. A different `vertical_id` may mean that the corresponding result has a different presentation style or comes from a different source. Organic results have a set of special `vertical_ids`, so we can also use them to separate organic results from vertical results.

We use two datasets, Dataset-C for the click prediction task (Section 4.4) and Dataset-R for relevance estimation task (Section 4.5), because relevance annotations are needed for the latter

[3]The *exploration bias* found by Chen et al. [4] assumes that after the user clicks a vertical result, she may choose not to click any organic results, which is similar to "satisfaction after click."

Table 2. The Statistics of Two Datasets Used in This Study

	#unique queries	#sessions	#unique URLs	#unique vertical_ids
Dataset-C	2,254,308	4,197,830	14,025,852	2,590
Dataset-R	546	21,443	2,742	171

task. In the relevance estimation task, we will train all click models on the training set of Dataset-C and evaluate them on Dataset-R according to the collected relevance annotations. The detailed statistics for the two datasets are shown in Table 2. We remove the sessions in the search logs without any click.

Dataset-C was generated by sampling about 3% sessions from five weekdays of one week. In the click prediction task, we divide Dataset-C into two subsets according to their time, with the first three-day data for training and the last two-day data for test.

Dataset-R was generated through the following process: (1) We first randomly sampled 12K unique queries from a one-month search log; (2) We then sampled all the sessions associated with those queries in the training set of Data-C, which may only cover 546 queries of those 12K queries; (3) We also crawled the SERPs for these queries and collected relevance labels for the top-five results of these queries using crowdsourcing. Because we assume that a user can be satisfied directly on the SERP, besides collecting relevance labels for the landing pages (Rel_{page}), we also collect relevance labels for the snippets ($Rel_{snippet}$) of mobile results. For the two kinds of relevance labels, we use two different sets of workers to make relevance judgments. During the judgment process, the system randomly selected a query and a snapshot of the snippet (or a web page) and showed them to the crowdsourcing workers at one time. After workers submitted the relevance judgment, the system repeated the above actions. A 4-level scale (1: not relevant, 2: somewhat relevant, 3: fairly relevant, and 4: perfectly relevant) was used for both Rel_{page} and $Rel_{snippet}$. Each snippet and landing page was annotated by at least three crowdsourcing workers and their relevance labels were determined by the median of all relevance annotations. All workers are Chinese with basic reading and computer skills. The values of Fleiss' κ [10] for Rel_{page} and $Rel_{snippet}$ are 0.75 and 0.73, both reaching a substantial agreement level (0–0.2: slight agreement; 0.2–0.4: fair agreement; 0.4–0.6: moderate agreement; 0.6–0.8: substantial agreement; 0.8–1.0: almost perfect agreement [24]). We also report the values of Krippendorf's alpha [21], which are 0.76 and 0.73 for Rel_{page} and $Rel_{snippet}$, respectively. All the values of Fleiss' κ and Krippendorf's alpha indicate an acceptable quality for relevance annotations.

4.1.2 Viewport Time. Although the viewport time can be treated as user attention, it is not only biased by the content of the result, but also by the presentation style, result heights, and other factors. On the one hand, different types of results usually have different heights with a fixed width in the mobile search environment. On the other hand, a result occupying more area in the viewport will naturally attract more user attention, leading to a longer viewport time. Lagun et al. [22] found that the *weighted viewport time* has the strongest correlation with user attention among all their methods. We follow them and adopt the *weighted viewport time* in our experiment to reduce the presentation bias of the viewport time. For a result in the SERP, its weighted viewport time is given as follows:

$$t_{weighted} = \sum_{i=1}^{n} t^i_{raw} * \frac{(h^i_e)^2}{h^i_v * h^i_r}, \tag{41}$$

where $t_{weighted}$ is the weighted viewport time of a result, n is the number of viewports in the query session, t^i_{raw} is the raw viewport time of the result in the ith viewport, h^i_e is the visible height of

the result exposed in the ith viewport, h_r^i is the actual height of the result, h_v^i is the height of the ith viewport. Specifically, h_e^i/h_v^i represents how much the result occupies the viewport (viewport coverage) and h_e^i/h_r^i represents how much the result is visible to the user (result exposure) [22]. About 4.3% of results' weighted viewport time is more than 30s in our dataset. To avoid the impact of abnormal data to our experiment, we follow Reference [39] and set 30s as an upper bound, correcting the abnormal viewport durations, which exceed the upper bound to 30s.

4.1.3 Baseline Models. We use three *basic* click models that do not take the type of search results into consideration, two *vertical-aware* click models originated from desktop search, and a *neural* click model as the baseline models. We refer to Chuklin et al. [7] for the implementations of the baseline models and make some necessary modifications to adapt them for a fair comparison on our dataset.

The three basic click models are the following:

- UBM: User Browsing Model proposed by Dupret and Piwowarski [9] (see Equations (2)–(4) in Section 2.2).
- DBN: Dynamic Bayesian Network model proposed by Chapelle and Zhang [3] (Equations (5)–(10)).
- DCM: Dependent Click Model proposed by Guo et al. [12].

Two vertical-aware baseline models are:

- EB-UBM: UBM with the exploration bias modification proposed by Chen et al. [4].
- UBM-layout: UBM that has different $\gamma_{i,d}$ parameters for search results with different layouts (i.e., different types of results). This model was designed by Chuklin et al. [7]. Note that here we cannot use the UBM-IA model proposed in the same article, because we do not have a pre-define classification of the search intent for each query. The original UBM-layout model only considers two types of results: *fresh* (i.e., news verticals) and *web* (i.e., organic results). We modify the model to make it work with an arbitrary number of result types defined by the `vertical_ids`. This modification improves UBM-layout model's performance in click prediction and relevance estimation. Therefore, we only report the performance of the modified UBM-layout model in this article.

The neural baseline model is:

- NCM: Neural Click Model proposed by Borisov et al. [1]. We implement NCM with the LSTM configuration.

4.2 Investigating Mobile Search Logs

Williams et al. [40] looked into how vertical results affect user clicks and found that different types of knowledge graph answers affect user behavior on SERPs differently. In this section, we would like to follow them and examine the effects of several common result types in the mobile search on user behaviors based on our dataset to support the designs of our mobile click models.

4.2.1 Comparison between Mobile and Desktop. Before training click models on the mobile search logs, we want to empirically demonstrate the differences between user click behavior in mobile and desktop search. So, we randomly sampled 10K mobile search sessions from Dataset-C and 10K desktop search sessions from the same commercial search engine to conduct a comparison analysis.

We first show the ratios of vertical results among the top 1, 3, 5, and 10 results in Figure 4. We can see that the ratios of vertical results in mobile search are higher than those in desktop search

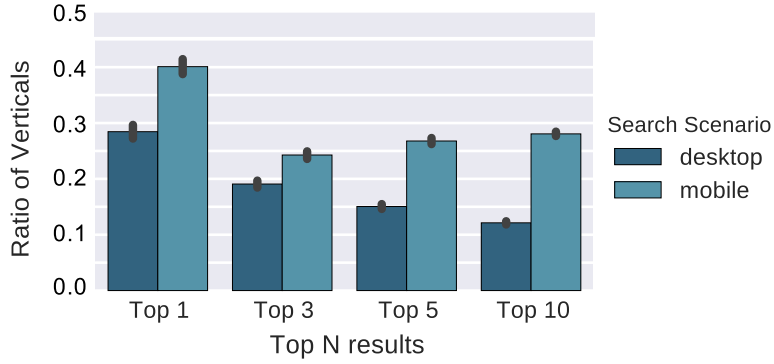


Fig. 4. The distribution of vertical results on desktop and mobile.

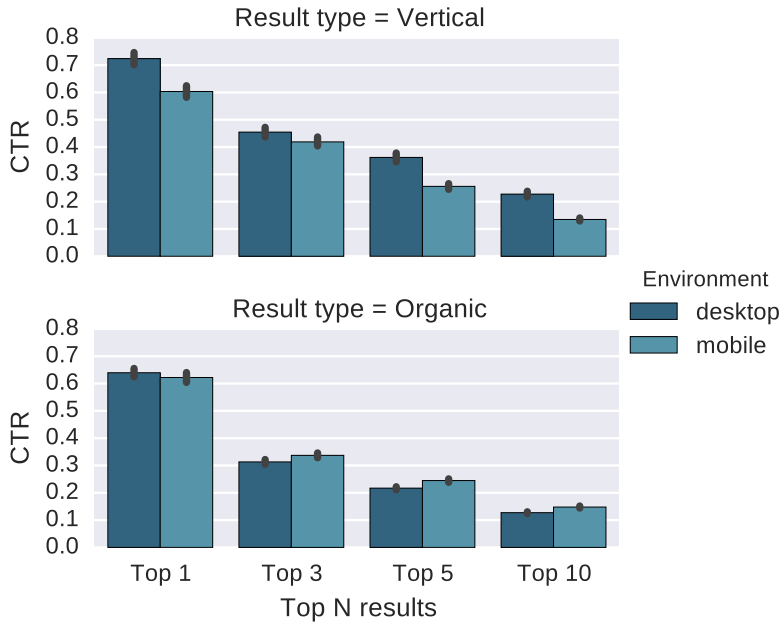


Fig. 5. The Click Through Rate (CTR) of vertical and organic results on desktop and mobile.

(all the differences are statistically significant at $p < 0.01$ level, independent t-test, two-tailed). On mobile SERPs, 28.1% of top-10 results and over 40% of the first search results are vertical, showing a prevalence of heterogeneous results in mobile search.

The click-through rates (CTRs) of both vertical and organic results are shown in Figure 5. For organic results, the click-through rates on mobile and desktop are comparable. For top-1 results, the click-through rates are not significantly different ($p = 0.14$). For top-3, top-5, and top-10 results, the click-through rates on mobile are slightly but significantly (all $p < 0.01$) higher than those on desktop with the absolute differences of 2.3%, 2.8%, and 2.0%, respectively. However, for vertical results, the click-through rates in mobile search are significantly lower than (all $p < 0.01$) those in desktop search with relatively large margins of 12.0%, 3.5%, 10.5%, and 8.3% for top-1, top-3, top-5, and top-10 results, respectively. The differences in click-through rates on vertical results in mobile

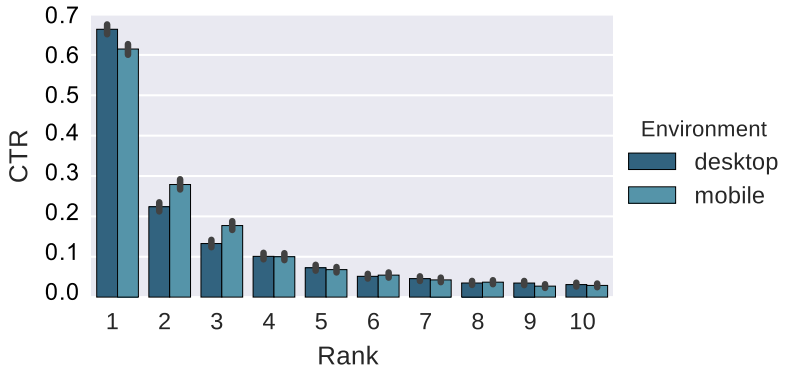


Fig. 6. The Click Through Rate (CTR) of top-10 results on desktop and mobile.

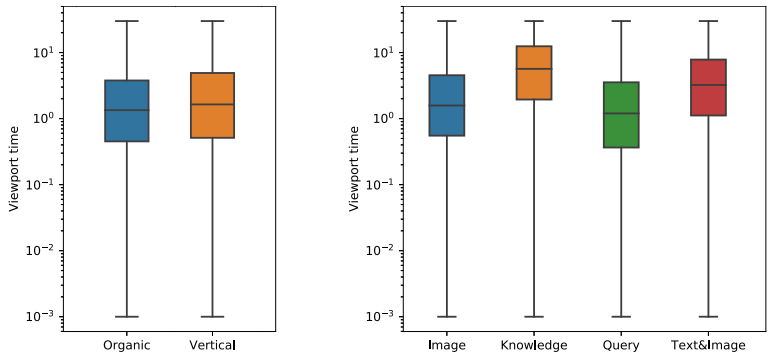
and desktop search imply that a large number of vertical results on mobile SERPs are designed to directly satisfy users without being clicked, which emphasizes the importance of modeling the click necessity bias in mobile context.

We also compare the position biases of click-through rates on mobile and desktop. From Figure 6, we can see that: (1) The click-through rate for the first mobile results is lower than that for the first desktop results, which can be explained by the fact that over 40% of the first mobile results are vertical and a large proportion of them can satisfy users without clicks. (2) The click-through rates for the second and third results on mobile are higher than those on the desktop. This finding is consistent with Lagun et al. [22] finding that mobile users tend to have a longer gaze time for the second and third results [22]. It can also be explained by the *spill-over* effect [26] that a user will pay more attention to the results below a visually attractive vertical result.

4.2.2 Viewport Time of Heterogeneous Results. To see if the types of results have an influence on the distributions of their viewport time, we conduct a statistic analysis of our dataset. For organic results, the results with zero viewport time account for 52.45%, while this percentage for vertical results is 49.36%. There also exist a few results with a rather long viewport time, so we artificially set the upper limit for 30s. There are only about 0.3% of results whose viewport time is longer than 30s in our search log dataset. For these results, we set their viewport time to 30s to reduce the impact of individual extreme values on the update of click model parameters. Figure 7 shows the positive viewport time distributions of heterogeneous results in our search log dataset, where we only use the results whose viewport time is longer than 0s. In Figure 7(a), we can see that the positive viewport time distribution of organic results (Mean = 2.87, IQR = [0.41, 3.74], SD = 3.99) is significantly different from that of vertical results (Mean = 3.79, IQR = [0.56, 4.89], SD = 5.16) at $p < 0.01$ level. For organic results, the mean viewport time is shorter than that of vertical results, and the standard deviation is smaller than that of vertical results. Figure 7(b) shows the positive viewport time distribution of four most-frequent vertical types of results in search logs. From this figure, we can see that viewport time distributions vary among different vertical types of results. Therefore, these results suggest we model the viewport time distribution for each type of results individually in VTCM.

4.3 Viewport Time Distribution Learned by VTCM

First, to answer **RQ1**, we would like to look into the selection problem of the viewport time distribution function in VTCM. We calculate the log-likelihood of different distribution functions with respect to the real viewport time data to see which distribution function can model the viewport



(a) Organic vs. vertical (b) Vertical

Fig. 7. The positive viewport time distributions of heterogeneous results in search logs, where only results with positive viewport time is included. Figure 7(a) shows the one of organic and vertical results. Figure 7(b) shows the one of four most-frequent vertical results. Image results contain several related images, while knowledge results provide a few question-answer pairs. "Query" means the query suggestion results. "Text&Image" represents results presented with a title, a textual snippet, and an image.

Table 3. Log-likelihood of Viewport Time Distribution Functions in VTCM*e* and VTCM*c*

Model	Log-normal	Gamma	Weibull
VTCM_e	−9.782	−8.981	−8.973
VTCM_c	−9.671	−8.961	**−8.957**

All differences are statistically significant at $p < 0.001$ level, pairwise t-test, two-tailed, $n = 1,910,040$.

time distributions best. The log-likelihood of a distribution function with respect to the real viewport time data is calculated as follows:

$$\begin{aligned} LL &= \frac{1}{|S|M} \sum_{s \in S} \sum_{i=1}^{M} log(P(V_i^s = t_i^s | C^s_{1,\ldots,M}, V^s_{1,\ldots,i-1,i+1,\ldots,M})) \\ &= \frac{1}{|S|M} \sum_{s \in S} \sum_{i=1}^{M} log\left(\sum_{\pi \in \Pi} P(\pi | C^s_{1,\ldots,M}, V^s_{1,\ldots,i-1,i+1,\ldots,M}) P(V_i = t_i | \pi) \right), \end{aligned} \tag{42}$$

$$\Pi_{VTCM_e} = \{E_i^s = 0, E_i^s = 1\}, \tag{43}$$

$$\begin{aligned} \Pi_{VTCM_c} = \{&E_i^s = 0, \\ &E_i^s = 1 \wedge C_i^s = 1 \wedge S_i^{E,s} = 0, \\ &E_i^s = 1 \wedge C_i^s = 0 \wedge S_i^{E,s} = 1, \\ &E_i^s = 1 \wedge C_i^s = 0 \wedge S_i^{E,s} = 0\}, \end{aligned} \tag{44}$$

where the viewport time t_i is treated as a discrete variable with the minimum unit 1ms for convenience of calculation. π is the condition of viewport time events V_i, while Π is the set of all possible π in the click model.

Table 3 shows the log-likelihood of three distribution functions applied in VTCM_e and VTCM_c. We can see that with the same distribution function, the log-likelihood of VTCM*c* is larger than

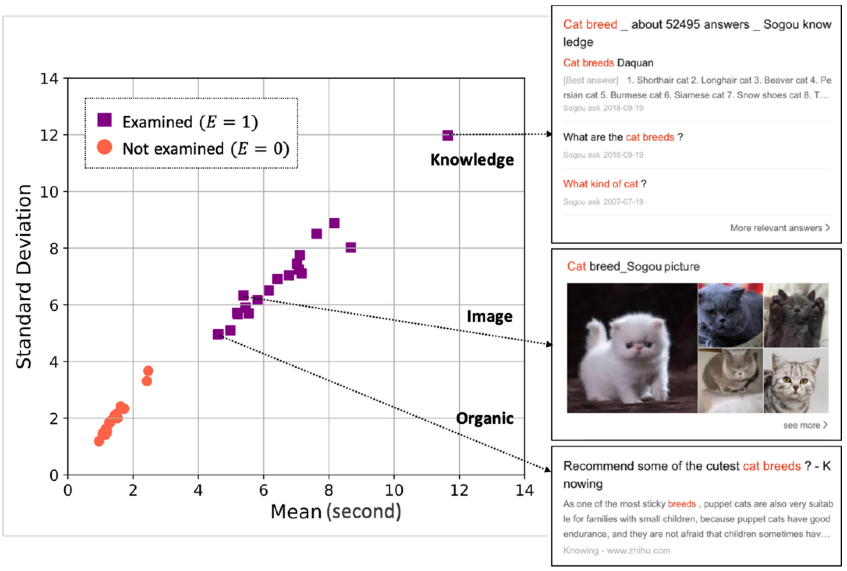


Fig. 8. The means and standard deviations of viewport time distributions learned by VTCM_e for the first 20 most-frequent types of results and three examples of results with different types. With the query "cat breed," the first knowledge vertical result contains several question-answer pairs relevant to the query, the second image vertical result provides images of cats, while the third organic result is a related web page.

that of VTCM_e, indicating that VTCM_c can model the viewport time distribution more effectively than VTCM_e. According to the performance of log-likelihood, Weibull is the best one among three viewport time distribution functions.

Figure 8 shows the means and standard deviations of viewport time distributions learned by VTCM_e for the first 20 most-frequent result types in our dataset. We can see that the mean viewport time for $E = 1$ (i.e., the result has been examined by the user) is larger than that for $E = 0$ (i.e., the result has not been examined), showing VTCM_e can effectively learn the difference of viewport time distributions under the two examination conditions. We select three representative cases: a knowledge vertical result, an image vertical result, and an organic result. The query of these snippets in Figure 8 is "cat breed" in English. The first snippet is a knowledge vertical result showing some related questions and answers written by other users. We can see that the mean viewport time of this result type for $E = 1$ is longer than that of other result types, indicating that users usually spend a rather long time on reading knowledge vertical results in mobile search. Meanwhile, we notice that its standard deviation is also larger than those of other vertical types. One possible reason is that the knowledge vertical results sometimes are not useful for some users with certain intent or not of high quality, so users prefer to spend a shorter time on viewing these results or directly ignore them. We also show another case, an image vertical result containing several images of cats, which users spend less time viewing than most other vertical types. The reason for this phenomenon may be complicated. We just list two possible reasons: On the one hand, it is naturally faster for users to get the overall meanings of images than text; on the other hand, due to the small display of mobile devices, users may tend to quickly click the image vertical results after they view it to see the target image clearly. We further look into the organic results. We can see that its mean viewport time is the shortest among all 20 result types, which can be caused by a lot of reasons, such as its low ranking position, low attractiveness, or relevance, and so on.

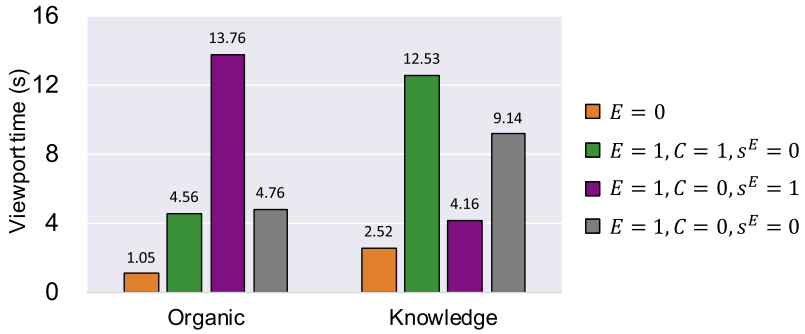


Fig. 9. The means of viewport time distributions of knowledge vertical results and organic results estimated by VTCM$_c$.

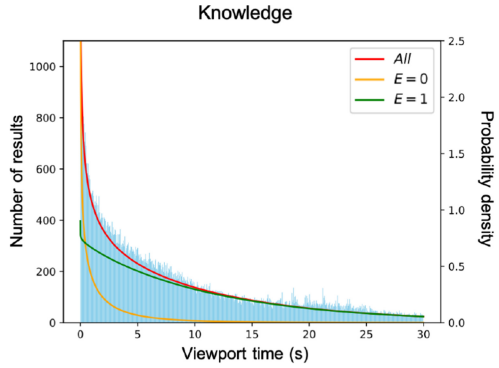
Figure 9 shows the means of viewport time distributions of knowledge vertical results and organic results learned by VTCM$_c$. A knowledge vertical result contains several question-answer pairs and occupies a large area in the viewport, while an organic result usually does not seem as attractive as a vertical result and occupies a relatively small area of the screen. Therefore, from the results, we can see that the mean viewport time of both knowledge vertical and organic results for $E = 0$ is shorter than that for the other three conditions. When the results have been clicked ($E = 1, C = 1, S^E = 0$) or the user just examines the result without any click and satisfaction ($E = 1, C = 0, S^E = 0$), the mean viewport time of knowledge vertical results is much longer than that of organic results, showing that users usually spend a long time on viewing knowledge vertical results than organic results. It usually takes users a shorter time to feel satisfied after examining the snippets of knowledge results compared to that of organic results ($E = 1, C = 0, S^E = 1$).

Figure 10 shows the real viewport time distributions of knowledge vertical results and organic results and the ones learned by VTCM*e* and VTCM*c*. To be compared with the real viewport time distributions, the learned distributions need to be weighted by their corresponding prior probabilities. For example, in VTCM*e*, the probability of V_i in a query session s is given as follows:

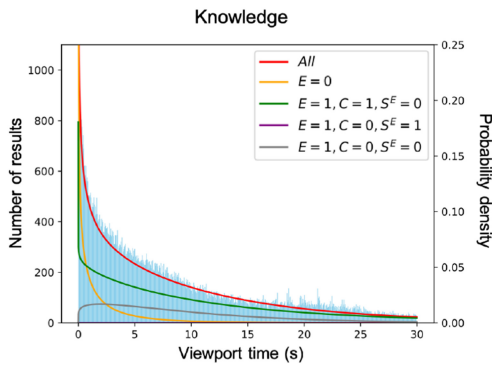
$$\begin{aligned} P(V_i = t_i | C^s_{1,\ldots,M}, V^s_{1,\ldots i-1,i+1,M}) = {} & P(E = 0 | C^s_{1,\ldots,M}, V^s_{1,\ldots i-1,i+1,M}) P(V_i = t_i | E = 0), \\ & + P(E = 1 | C^s_{1,\ldots,M}, V^s_{1,\ldots i-1,i+1,M}) P(V_i = t_i | E = 1), \end{aligned} \tag{45}$$

where $P(E = 0 | C^s_{1,\ldots,M}, V^s_{1,\ldots i-1,i+1,M})$ and $P(E = 1 | C^s_{1,\ldots,M}, V^s_{1,\ldots i-1,i+1,M})$ are prior probabilities, while $P(V_i = t_i | E = 0)$ and $P(V_i = t_i | E = 1)$ are called posterior probabilities. For the convenience of visualization, we use the mean of all prior probabilities under a condition to weight the corresponding viewport time distribution.

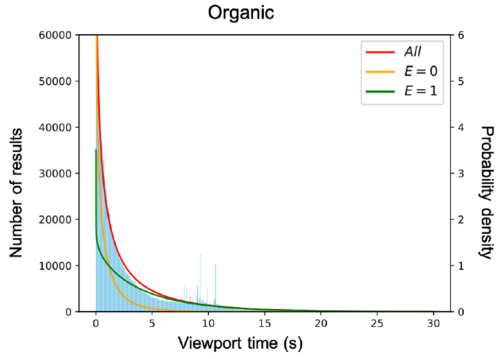
In Figure 10, the distribution "All" (the red line) is the sum of all distributions for possible conditions. From Figure 10, we can see that both VTCM*c* and VTCM*e* are capable of modeling the viewport time distributions of the two result types. The mean probabilities $P(E = 0, C = 0, S^E = 1 | C^s_{1,\ldots,M}, V^s_{1,\ldots i-1,i+1,M})$ for organic and knowledge results are, respectively, 4.37×10^{-3} and 3.14×10^{-7}, causing the purple lines to be very close to X-axis, which also indicates that users hardly leave (i.e., feel satisfied under the hypothesis of VTCM) after viewing these two types of results. We further look into the posterior probabilities estimated by VTCM$_c$. Figure 11 shows the ratio of posterior probabilities under four conditions with the increase of the viewport time. First, we can see as the viewport time increases, the ratio of the posterior probability for $E = 0$ decreases, indicating that in VTCM*c*, the longer viewport time a result has, the more likely the user has



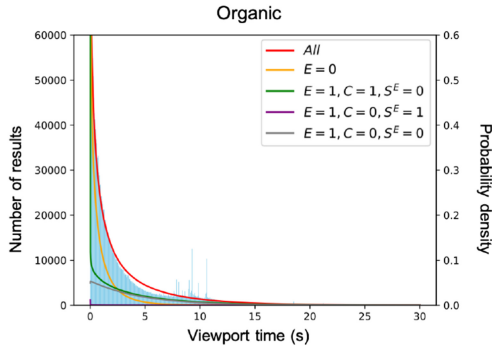
(a) Knowledge in VTCM$_e$



(b) Knowledge in VTCM$_c$



(c) Organic in VTCM$_e$



(d) Organic in VTCM$_c$

Fig. 10. The viewport time distributions of knowledge vertical results and organic results learned by VTCM$_e$ and VTCM$_c$.

examined it. Second, for knowledge vertical results, the ratio of $E = 0, C = 1, S^E = 0$ gets larger as the viewport time increases, which means that the longer time a user spends on viewing a knowledge vertical result, the more likely the user will click it. Third, for organic results, the ratio for $E = 1, C = 0, S^E = 1$ gets larger as the viewport time increases, which is mainly because the posterior probabilities for the other three conditions get much smaller at the same time, indicating that for an organic result with a rather long viewport time, it is less possible for users to click it.

Regarding as **RQ1**, we show how VTCM$_e$ and VTCM$_c$ effectively model the viewport time distributions of heterogeneous results in mobile search. We also draw from the experimental results that $VTCM_c$ can outperform $VTCM_e$ in both click prediction and relevance estimation tasks based on modeling viewport time under more fine-grained conditions.

4.4 Click Prediction

When measuring the click prediction performance on the test set, we filter out all the queries that have a query frequency less than 10 in the training set. Following the convention of previous works [4, 9, 38], we use two evaluation metrics, log-likelihood (*LL*) and average perplexity (*AvgPerp*), to evaluate models' performance in the click prediction task. The log-likelihood

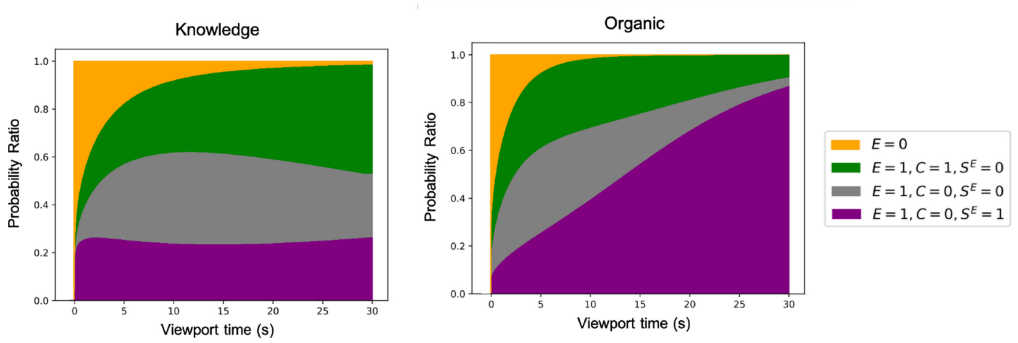


Fig. 11. The posterior probability ratio of four conditions for knowledge vertical results and organic results in VTCM$_c$.

functions of MCM and VTCM are calculated as follows:

$$LL_{MCM} = \frac{1}{|S|}\sum_{s\in S}\sum_{i=1}^{M} log(P(C_i^s|C_1^s,\ldots,C_{i-1}^s)) = \frac{1}{|S|}\sum_{s\in S} log(P(C_1^s,\ldots,C_M^s), \tag{46}$$

$$LL_{VTCM} = \frac{1}{|S|}\sum_{s\in S}\sum_{i=1}^{M} log(P(C_i^s|C_1^s,\ldots,C_{i-1}^s,V_1^s,\ldots,V_i^s)). \tag{47}$$

The average perplexity is the mean of perplexity $Perp_i$ over M positions:

$$AvgPerp = \frac{1}{M}\sum_{i=1}^{M} Perp_i, \tag{48}$$

$$Perp_i = 2^{-\frac{1}{|S|}\sum_{s\in S} C_i^s \log_2(q_i^s)+(1-C_i^s)\log_2(1-q_i^s)}, \tag{49}$$

where $M = 10$ and S is the set of all search sessions in test set, while q_i^s is the conditional click probability of result i in session $s \in S$, given the clicks of the first $i-1$ results, i.e., $P(C_i^s|C_1^s,\ldots,C_{i-1}^s)$, predicted by the baseline click models and MCM. For VTCMs, q_i^s represents $P(C_i^s|C_1^s,\ldots,C_{i-1}^s,V_1^s,\ldots,V_i^s)$. A larger LL indicates a better performance, and the relative improvement of LL_1 over LL_2 is given by $\exp(LL_1 - LL_2) - 1$). While the perfect prediction at position i will have a perplexity $Perp_i = 1.0$, a smaller value of $AvgPerp$ indicates better prediction accuracy. The relative improvement of perplexity value p_1 over p_2 is computed as $(p_2 - p_1)/(p_2 - 1)$.

4.4.1 Comparison among VTCMs with Different Viewport Time Distribution Functions. To answer **RQ1**, we also look into the overall click prediction performance of VTCMs with different distribution functions on the remaining 283,004 sessions of the test data, which is shown in Table 4. From the results, we can see that among three viewport time distribution functions, VTCMs with Weibull achieve the best performance in the click prediction task, followed by gamma, while VTCMs with log-normal perform worst. With the same viewport time distribution function, VTCM$_c$ has better click prediction ability than VTCM$_e$, indicating that modeling the relationships among viewport time, user clicks, and examination satisfaction can improve the click prediction performance.

Table 4. The Overall Click Prediction Performance of VTCMs with Different Viewport Time Distribution Functions, Which is Measured in Log-likelihood (*LL*) and Average Perplexity (*AvgPerp*)

Model	Viewport	*LL*	*SD* ($\times 10^{-6}$)	*AvgPerp*	*SD* ($\times 10^{-6}$)
$VTCM_e$	log-normal	−2.013	7.88	1.236	2.58
	gamma	−2.011	7.68	1.235	2.51
	Weibull	−1.997	7.71	1.233	2.51
$VTCM_c$	log-normal	−2.007	8.05	1.235	2.64
	gamma	−1.989	7.91	1.233	2.60
	Weibull	**−1.982**	7.94	**1.232**	2.60

All differences over the best results (which are bold) are statistically significant at $p < 0.001$ level, pairwise t-test, two-tailed, $n = 283,004$. *SD* is the standard deviation of the mean of *LL* and *AvgPerp*.

Table 5. The Overall Click Prediction Performance of VTCMs, MCM, and Baselines Measured in Log-likelihood (*LL*) and Average Perplexity (*AvgPerp*)

Model	*LL*	Impr.	*SD* ($\times 10^{-6}$)	*AvgPerp*	Impr.	*SD* ($\times 10^{-6}$)
DBN	−2.249	−5.27%	8.78	1.263	−5.29%	2.70
DCM	−2.233	−4.52%	7.71	1.263	−5.39%	2.21
UBM	−2.175	−1.82%	7.77	1.256	−2.27%	2.22
EB-UBM	−2.182	−2.12%	7.65	1.257	−2.65%	2.18
UBM-layout	−2.144	−0.35%	7.67	1.251	−0.52%	2.22
NCM	−2.131	0.26%	7.46	1.249	0.36%	2.14
MCM	−2.136	-	7.71	1.250	-	2.23
$VTCM_e$	−1.997	6.54%	7.71	1.233	6.68%	2.51
$VTCM_c$	**−1.982**	7.21%	7.94	**1.232**	7.18%	2.60

All relative improvements over MCM are calculated and statistically significant at $p < 0.001$ level, pairwise t-test, two-tailed, $n = 283,004$. *SD* is the standard deviation of the mean of *LL* and *AvgPerp*.

4.4.2 Comparison with Baselines. To answer **RQ2**, we compare the overall click prediction performance of MCM, VTCMs, and baselines in Table 5. Note that in the following experiment, we adopt Weibull distribution for both $VTCM_c$ and $VTCM_e$, because VTCMs with Weibull distribution can achieve the best performance in both modeling the viewport time distributions and predicting user clicks among all three distribution functions in our previous experiment. We can find in Table 5 that MCM significantly outperforms all PGM-based baseline models, including the *basic* and *vertical-aware* baselines, but performs worse than NCM, while VTCM significantly outperforms MCM and all baselines.

We further compare different models for queries with different query frequencies. From Table 6, we can see that while the click performance measured in *LL* increases as the query frequency for all models, $VTCM_c$ performs consistently the best among all click models, followed by $VTCM_e$ and MCM. It is interesting to see that the relative improvements of MCM over baseline models are larger for queries whose frequency is over 50 in the training set. A possible reason for this phenomenon is that there are more vertical results designed for and federated into the SERPs of hot queries. We compare the performances of MCM and NCM and find that with the increase of the query frequency, the performance of MCM gradually gets better than that of NCM. Compared to MCM, VTCMs achieve the largest relative improvements for queries whose frequency is below 50. A possible reason for this phenomenon is that when the query is of low frequency, the viewport

Table 6. Log-likelihood of Each Model for Different Query Frequency in Training Set

Query Freq.	[10, 50)		[50, 200)		[200, inf)	
#Sessions	152,265		78,187		52,552	
Model	*LL*	Impr.	*LL*	Impr.	*LL*	Impr.
DBN	−2.503	−5.74%	−2.118	−4.72%	−1.706	−4.30%
DCM	−2.436	−2.91%	−2.133	−5.45%	−1.792	−9.56%
UBM	−2.396	−1.19%	−2.068	−2.23%	−1.696	−3.68%
EB-UBM	−2.395	−1.16%	−2.078	−2.73%	−1.718	−5.01%
UBM-layout	−2.356	0.47%	−2.047	−1.20%	−1.673	−2.24%
NCM	−2.346	0.89%	−2.036	−0.65%	−1.648	−0.73%
MCM	−2.367	-	−2.023	-	−1.636	-
$VTCM_e$	−2.173	8.21%	−1.915	5.34%	−1.607	1.78%
$VTCM_c$	**−2.161**	8.70%	**−1.896**	6.27%	**−1.592**	2.69%

All relative improvements over MCM are calculated and statistically significant at $p < 0.001$ level, pairwise t-test, two-tailed, n = #Sessions.

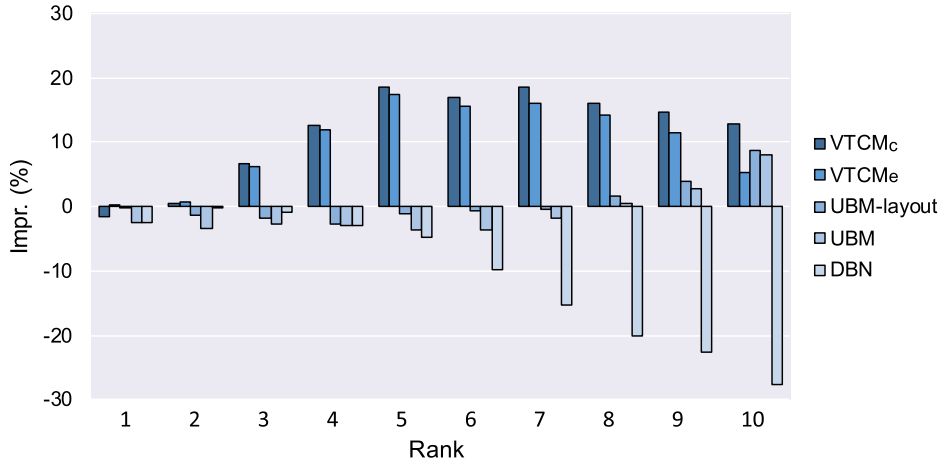


Fig. 12. Perplexity improvements of click models at different ranking positions compared to MCM. All improvements over MCM are statistically significant at $p < 0.001$ level, pairwise t-test, two-tailed, $n = 283,004$.

time information can serve as a kind of users' positive feedback and help VTCM effectively estimate the examination probability of results as well as the parameters of examination satisfaction. We will discuss how VTCM learns parameters with the viewport time information in Section 4.6.

We are also interested in the prediction performance at each ranking position. In Figure 12, we plot the relative perplexity improvements of MCM over two basic baseline models, UBM and DBN, one best performing vertical-aware baseline, UBM-layout. It is worth noting that two basic click models, UBM and DBN, behave differently in mobile search. While all the models have comparable performance at position 1, MCM has larger gains over UBM for positions 2–4 and over DBN for positions 5–10. UBM performs worse at positions 2–4, because it cannot adjust the examination probability accordingly when some top-ranked results already satisfy the user. DBN's performance drops as the rank increases, because the skip examination behavior is more common at the lower

positions, which violates DBN's assumption. MCM overcomes these disadvantages by incorporating examination/click satisfaction and allowing skip examination. Therefore, it has a consistent improvement over UBM and DBN at positions 1–7. We speculate that MCM is worse than UBM at positions 8–10 because the irregularity of click-through rate at the lower positions can be easily captured by UBM, but for MCM, the estimation of examination probability at position i ($P(E_i = 1)$) is dominated by the satisfaction probability ($P(S_{i-1} = 1)$). The vertical-aware UBM-layout model has a similar performance patterns to UBM. Because UBM-layout can capture the attention bias on examination probability, it consistently outperforms UBM across 10 positions.

In Figure 12, we also plot the relative perplexity improvements of VTCM$_c$ and VTCM$_e$ over MCM. We can find that the relative improvements (or deteriorations) of VTCMs over MCM are smaller at positions 1 and 2 than that at other positions. VTCM$_e$ slightly outperforms MCM at the top-two positions, while VTCM$_c$ consistently has larger improvements over MCM than VTCM$_e$ at positions 3–10. One reason for the limited positive and even negative improvements of VTCMs over MCM at positions 1 and 2 is that the top-two results in the SERP usually have longer viewport time than results at lower positions, so the overall distributions learned for all positions may not work, which inspires us to model the distributions of each position individually, and we would like to leave it as a future work. From positions 3 to 5, the performance gaps between VTCM models and MCM get larger, while it becomes relatively smaller from positions 7–10. To sum up, with the help of viewport time information, VTCMs can achieve better click prediction performance than MCM at those positions where user feedbacks are rare.

Regarding **RQ1**, we find that Weibull distribution is the best viewport time distribution function in our experiment according to its outstanding performance in the click prediction task. Regarding **RQ2**, we find that VTCM$_c$ has the best click prediction ability in the mobile search environment among all the click models, followed by VTCM$_e$, NCM, and MCM. The improvements of VTCMs and MCM are consistent for queries with different frequencies and for almost all positions in the first page. These results suggest that incorporating the click necessity bias, examination satisfaction bias, and viewport time information is effective in modeling user click behavior in mobile search.

4.5 Relevance Estimation

To address **RQ3**, we evaluate MCM, VTCM$_e$, VTCMc, and baseline models on Dataset-R and rank the results according to the predicted relevance score provided by each model. Besides $Rel_{snippet}$ and Rel_{page}, we also compute an average of these two labels $Rel_{avg} = (Rel_{page} + Rel_{snippet})/2$ [4]. The ranking results can be evaluated by standard IR evaluation metrics. In this study, we use $nDCG@\{1, 3, 5\}$ [16], $nERR@5$ [2], and $MAP@5$ [34] as the evaluation metrics for the relevance estimation task.

Table 7 shows the ranking performance of each click model based on three kinds of relevance labels. From the results, we can see that the vertical-aware models are generally better than the basic models in relevance estimation. These results emphasize the importance of considering the heterogeneity of search results in the mobile context. MCM has better performance than the two vertical-aware baselines, because the click necessity bias may have a stronger influence on user click behavior than the attention bias in the mobile environment. VTCM models have significant improvement over MCM in the performance of relevance estimation, as they can capture user attention during the search process with the information of viewport time. For NCM, we follow Borisov et al. [1] and adopt $P(C_1 = 1 | q, d)$, the click probability of a document when it appears

[4]We also tested the geometric and harmonic mean. The results were similar to the arithmetic mean, so we only report the results using arithmetic mean in the article.

Table 7. Relevance Estimation Performance of Click Models Based on $Rel_{snippet}$, Rel_{page}, and Rel_{avg} as well as Relative Improvements Over MCM

$Rel_{snippet}$										
Model	*nDCG*@1		*nDCG*@3		*nDCG*@5		*nERR*@5		*MAP*@5	
DBN	0.762**	−5.85%	0.774**	−3.47%	0.859**	−1.57%	0.874**	−2.54%	0.850**	−2.03%
DCM	0.805	−0.58%	0.790**	−1.46%	0.867	−0.62%	0.893	−0.36%	0.861**	−0.71%
UBM	0.789	−2.61%	0.781**	−2.60%	0.865*	−0.94%	0.885*	−1.28%	0.855**	−1.44%
EB-UBM	0.791	−2.38%	0.783**	−2.34%	0.866	−0.83%	0.887	−1.09%	0.856**	−1.34%
UBM-layout	0.786*	−2.91%	0.778**	−2.94%	0.860**	−1.41%	0.883*	−1.54%	0.847**	−2.33%
NCM	0.732**	−9.56%	0.745**	−7.09%	0.845**	−3.16%	0.850**	−5.23%	0.817**	−5.83%
MCM	0.810	-	0.802	-	0.873	-	0.897	-	0.867	-
$VTCM_e$	0.852**	5.21%	0.828**	3.31%	0.888**	1.77%	0.917**	2.23%	0.883**	1.86%
$VTCM_c$	**0.855****	5.59%	**0.836****	4.27%	**0.891****	2.12%	**0.919****	2.52%	**0.885****	2.10%
Rel_{page}										
Model	*nDCG*@1		*nDCG*@3		*nDCG*@5		*nERR*@5		*MAP*@5	
DBN	0.809**	−4.76%	0.792**	−2.42%	0.869**	−1.01%	0.894**	−1.59%	0.850**	−2.18%
DCM	0.793**	−6.60%	0.784**	−3.41%	0.864**	−1.62%	0.891**	−1.96%	0.850**	−2.18%
UBM	0.818**	−3.61%	0.791**	−2.50%	0.871**	−0.80%	0.898**	−1.24%	0.852**	−2.02%
EB-UBM	0.821**	−3.31%	0.795**	−2.12%	0.873**	−0.62%	0.900**	−0.99%	0.852**	−1.94%
UBM-layout	0.828**	−2.50%	0.796**	−1.92%	0.872**	−0.62%	0.902**	−0.76%	0.850**	−2.18%
NCM	0.769**	−9.38%	0.760**	−6.34%	0.856**	−2.47%	0.872**	−4.12%	0.822**	−5.43%
MCM	0.849	-	0.812	-	0.878	-	0.909	-	0.869	-
$VTCM_e$	0.857*	0.88%	0.827**	1.89%	0.887**	1.08%	0.921**	1.36%	0.878**	1.07%
$VTCM_c$	**0.860***	1.32%	**0.835****	2.84%	**0.891****	1.46%	**0.926****	1.83%	**0.882****	1.45%
Rel_{avg}										
Model	*nDCG*@1		*nDCG*@3		*nDCG*@5		*nERR*@5		*MAP*@5	
DBN	0.763**	−5.52%	0.766**	−3.72%	0.856**	−1.62%	0.871**	−2.88%	0.880**	−1.96%
DCM	0.776	−3.97%	0.771**	−3.09%	0.857**	−1.52%	0.881*	−1.81%	0.883**	−1.53%
UBM	0.781*	−3.28%	0.770**	−3.23%	0.860**	−1.17%	0.880**	−1.90%	0.881**	−1.76%
EB-UBM	0.785*	−2.80%	0.773**	−2.89%	0.861**	−1.01%	0.882**	−1.68%	0.882**	−1.69%
UBM-layout	0.787	−2.53%	0.772**	−2.94%	0.859**	−1.24%	0.882**	−1.70%	0.875**	−2.42%
NCM	0.737**	−8.73%	0.740**	−6.98%	0.845**	−2.92%	0.853**	−4.92%	0.846**	−5.67%
MCM	0.808	-	0.796	-	0.870	-	0.897	-	0.897	-
$VTCM_e$	0.832*	3.01%	0.812*	2.06%	0.880**	1.13%	0.909**	1.29%	0.909**	1.30%
$VTCM_c$	**0.837***	3.59%	**0.821****	3.15%	**0.883****	1.51%	**0.913****	1.77%	**0.912****	1.62%

*/** indicates the difference over MCM is significant at $p < 0.05/0.01$ level, pairwise t-test, two-tailed, $n = 546$.

in the first position, as its estimated relevance. We can find that the performance of NCM is the worst among all models in the relevance estimation task. We attribute this phenomenon to the fact that NCM was not designed for relevance estimation. The click probability is not suitable to serve as relevance, because it is not only related to the query and the document but also affected by the examination behavior of users. In the mobile search environment, we hold that the effect of examination behavior on the click probability is stronger because of the click necessity bias and examination satisfaction bias.

To sum up, regarding **RQ2**, we show that MCM has a better performance in estimating the relevance of mobile search results than the baseline models, although some differences are not significant at $p < 0.05$ level. Both $VTCM_e$ and $VTCM_c$ have significantly better performances in

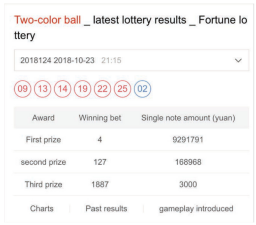
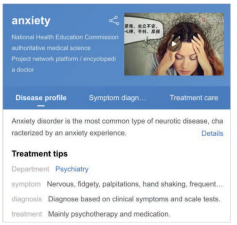
Snippet			
Query	双色球 (A popular lottery in China)	wifi修改密码 (Changing Wi-Fi password)	焦虑症 (Anxiety neurosis)
Result	A direct answer result showing latest result of the lottery and the prizes.	A vertical result showing the step-by-step guidance of how to changing the Wi-Fi password.	A medical knowledge graph result showing the symptom, diagnosis and treatment of anxiety neurosis.
Parameters	$\beta = 0.0002, s^E = 0.860$	$\beta = 0.0002, s^E = 0.502$	$\beta = 0.0003, s^E = 0.217$

Fig. 13. Examples of search results that have the lowest click necessity estimated by MCM (β).

relevance estimation than MCM and all the baseline models. We can also see that VTCM$_c$ is consistently better than VTCM$_e$, but their differences are not significant.

4.6 Parameter Analysis

To see how MCM models the click necessity bias and examination satisfaction bias in mobile search (**RQ4**), we analyze the click necessity parameters β and examination satisfaction parameters s^E learned by MCM.

We first compute the mean of click necessity parameter β of MCM for all organic results and vertical results. Mean $\beta_{vertical}$ on Dataset-C is 0.832 ($SD = 0.194$), which is significantly lower than $\beta_{organic} = 0.953$. This confirms our observation in Section 4.2.1 that the vertical results in mobile search are more likely to have a low click necessity. However, the mean of examination satisfaction parameter s^E for the vertical results $M(s^E_{vertical})$ is 0.242($SD = 0.107$), fairly close to $M(s^E_{organic})$, which equals to 0.244 ($SD = 0.108$). This result suggests that only a small proportion of vertical results can directly lead to examination satisfaction.

We further conduct a case study to inspect the relationship between the model parameters in MCM and the search results. Three types of vertical results with lowest β are shown in Figure 13. For each vertical type, we select a result and the corresponding query from the logs as an example.[5] We can see that these examples all demonstrate useful information directly in the snippet. A user can get information from them without a click, which is captured by the β parameter. We also show the estimation of s^E for each query-result pair. We can see that the learned s^E can reflect the examination satisfaction to some extent. Users are likely to be satisfied by the result in the first example if they just want to know the latest lottery result. This can be reflected by a high s^E of 0.860. However, if a user wants to get sufficient information about *anxiety neurosis*, although the medical knowledge graph result in the last example can provide a good overview, it is less likely for the user to be satisfied by this single result. Therefore, the corresponding s^E estimated by MCM is only 0.217.

We acknowledge that these examples also reveal a limitation of MCM, which is that s^E may not be a valid relevance indicator in the complex, informational tasks. Merely incorporating the

[5]The snippet of result is crawled by us, which may be different from the result viewed by the user in the search log, even if they share the same URL and vertical_id.

Table 8. An Example of a Query Session with the Viewport Time, User Clicks, and Related Probabilities Learned by $VTCM_c$ and MCM, Where the Query Is "avatar" in English

Rank (i)	Viewport time (s)	Click	Examination prob.		Satisfaction prob.		Global satisfaction prob.	
			$VTCM_c$	MCM	$VTCM_c$	MCM	$VTCM_c$	MCM
1	18.60	1	1	1	0	0	0	0
2	7.61	0	0.970	0.755	0	0	0	0
3	8.15	0	0.987	0.186	0	0	0	0
4	11.90	0	0.968	0.159	0	0	0	0
5	9.71	1	1	1	0.556	0.353	0.556	0.353
6	0	0	0	0.138	0	0.002	0.556	0.354
7	0	0	0	0.177	0	0.010	0.556	0.360
8	0	0	0	0.108	0	0.002	0.556	0.362
9	0	0	0	0.056	0	0.003	0.556	0.364
10	0	0	0	0.052	0	0.001	0.556	0.364

examination satisfaction bias cannot fully solve the problem of assigning positive feedback to the results that were not clicked.

We incorporate the viewport time information into VTCM as kind of a positive feedback from users. Thus, to investigate how VTCM estimates the parameters of examination and satisfaction with the help of viewport time information, we show a session case of the query "avatar" in our dataset, including the viewport time and user clicks of the top-10 results in the first page as well as related probabilities learned by $VTCM_c$ and MCM, which is shown in Table 8. First, we give the mathematical expressions of probabilities shown in Table 8. For VTCM, the examination probability at ranking position i in the session s is $P(E_i^s = 1|C^s, V^s)$ and the satisfaction probability is $P(E_i^s = 1, S_i^s = 1|C^s, V^s)$, i.e., the probability that the user is unsatisfied after interacting with the first $i-1$ results and becomes satisfied after viewing the ith result. The global satisfaction for VTCM is $P(S_i^s = 1|C^s, V^s)$, which means the probability that the user feels satisfied with one of the first i results. For MCM, the three probabilities are $P(E_i^s = 1|C^s)$, $P(E_i^s = 1, S_i^s = 1|C^s)$, and $P(S_i^s = 1|C^s)$, respectively. When comparing the estimated examination probabilities, we can see that it is more likely that the user has examined the top-five results because of the rather long viewport time and two user clicks. All examination probabilities of top-five results learned by VTCM are more than 0.9, while the examination probabilities of the third and fourth results learned by MCM are below 0.2. At those positions whose results have zero viewport time, the examination probabilities estimated by VTCM are zero, while those estimated by MCM are larger than zero. As for the satisfaction probabilities, we can see that the user left after examining the fifth result. Thus, it is because this result made the user satisfied based on the hypotheses of VTCM and MCM. From the results, we can see that VTCM estimates a higher satisfaction probability for the fifth result than MCM. We also calculate the global satisfaction probability for each position. We can see that after position 5, this probability given by VTCM is higher than that given by MCM. All these results show the effectiveness of VTCM in estimating the probabilities and parameters of examination and satisfaction over MCM.

5 CONCLUSIONS AND FUTURE WORK

Observing that in mobile search, some vertical results (such as direct answer results and knowledge graph results) have low click necessity and, therefore, will be discriminated by most existing click models, we propose a simple yet effective Mobile Click Model (MCM) to incorporate the related click necessity bias and examination satisfaction bias in mobile search. Theoretically, the proposed

MCM extends the examination hypothesis and can be regarded as a unified generalization of two most widely used click models, DBN and UBM. Meanwhile, we show that we can utilize viewport time to further calibrate the estimated probabilities of user examination and satisfaction. We extend MCM by incorporating viewport time bias into it as the second observed variable and propose Viewport Time Click Model (VTCM). Specifically, VTCM models the viewport time distributions for each type of results under different conditions of user examination, user clicks, and examination satisfaction. Empirically, extensive experiments on large-scale mobile search logs demonstrate that MCM and VTCM outperform the baseline models in both the click prediction task and relevance estimation task.

In terms of future work, we note that MCM and VTCM can be further extended in many ways. First, while the click necessity parameters β are fully learned from click logs in this study, we can introduce external knowledge into MCM to further improve its effectiveness by defining the prior of β for each type of vertical result accordingly. Second, as we mentioned in Section 4.6, the current definition of examination satisfaction may fail to reflect the relevance of results in complex search tasks. Instead of always attributing satisfaction to the last-clicked or last-examined result, we can explore new ways to properly measure the contribution of every search result. Third, as we mentioned in Section 4.4.2, the viewport time distributions may vary among different ranking positions. Thus, we can try to model the viewport time distributions for different ranking positions individually in the click model, which may bring in more improvements. It is also worth noting that while this study is motivated by the difference between mobile and desktop search, the click necessity bias and examination satisfaction bias may exist in desktop search, too. In future work, we will try to adopt MCM and VTCM to the desktop environment, where the mouse events are possible to be used to improve the satisfaction estimations of vertical results whose click necessity is low.

APPENDIX

We publicly released the implementations of our mobile click models, including MCM and VTCM, to contribute to the research community.[6] In the Appendix, we will first introduce how to update the parameters of MCM $\{\alpha, \beta, \gamma, s^E, s^C\}$ and the viewport time-related parameters of VTCM Θ in the M-step using the posterior distributions of the latent variables $\{E_i, A_i, N_i, S_i^E, S_i^C, S_i\}$. After that, we will give some details about how to compute the posterior distributions using a forward-backward algorithm.

M-step

Suppose we have a set of search sessions S and each session $s \in S$ is associated with a query q^s and M search results $(d_1^s, \ldots, d_M^s)$ with types $(v_1^s, \ldots, v_M^s)$. We denote $E^s, A^s, N^s, S^{E,s}, S^{C,s}$ the vector of latent variables in a session s. The updates of the parameters $\{\alpha, \beta, \gamma, s^E, s^C\}$ in MCM are as follows:

$$\alpha_{q,d} = \operatorname*{argmax}_{\alpha} \sum_{s \in S} \sum_{i=1}^{M} I(q^s = q, d_i^s = d) \\ [P(A_i^s = 1|C^s) \log(\alpha) + P(A_i^s = 0|C^s) \log(1 - \alpha)],$$

$$\beta_v = \operatorname*{argmax}_{\beta} \sum_{s \in S} \sum_{i=1}^{M} I(v_i^s = v) \\ [P(N_i^s = 1|C^s) \log(\beta) + P(N_i^s = 0|C^s) \log(1 - \beta)],$$

[6] https://github.com/THUIR/click_model_for_mobile_search.

$$\gamma_{r,d} = \operatorname*{argmax}_{\gamma} \sum_{s\in S}\sum_{i=1}^{M} I(i = r, i - Pos(lastClick, i) = d)$$
$$[P(E_i^s = 1, S_{i-1}^s = 0|C^s)\log(\gamma) + P(E_i^s = 0, S_{i-1}^s = 0|C^s)\log(1-\gamma)],$$

$$s_{q,d}^C = \operatorname*{argmax}_{s^C} \sum_{s\in S}\sum_{i=1}^{M} I(q^s = q, d_i^s = d)$$
$$[P(S_i^{C,s} = 1, C_i^s = 1|C^s)\log(s^C) + P(S_i^{C,s} = 0, C_i^s = 1|C^s)\log(1-s^C)],$$

$$s_{q,d}^E = \operatorname*{argmax}_{s^E} \sum_{s\in S}\sum_{i=1}^{M} I(q^s = q, d_i^s = d)$$
$$[P(S_i^{E,s} = 1, E_i^s = 1, A_i^s = 1, N_i^s = 0|C^s)\log(s^E)$$
$$+ P(S_i^{E,s} = 0, E_i^s = 1, A_i^s = 1, N_i^s = 0|C^s)\log(1-s^E)].$$

In VTCM, the updates of the parameters $\{\alpha, \beta, \gamma, s^E, s^C\}$ as follows:

$$\alpha_{q,d} = \operatorname*{argmax}_{\alpha} \sum_{s\in S}\sum_{i=1}^{M} I(q^s = q, d_i^s = d)$$
$$[P(A_i^s = 1|C^s, V^s)\log(\alpha) + P(A_i^s = 0|C^s, V^s)\log(1-\alpha)],$$

$$\beta_v = \operatorname*{argmax}_{\beta} \sum_{s\in S}\sum_{i=1}^{M} I(v_i^s = v)$$
$$[P(N_i^s = 1|C^s, V^s)\log(\beta) + P(N_i^s = 0|C^s, V^s)\log(1-\beta)],$$

$$\gamma_{r,d} = \operatorname*{argmax}_{\gamma} \sum_{s\in S}\sum_{i=1}^{M} I(i = r, i - Pos(lastClick, i) = d)$$
$$[P(E_i^s = 1, S_{i-1}^s = 0|C^s, V^s)\log(\gamma) + P(E_i^s = 0, S_{i-1}^s = 0|C^s, V^s)\log(1-\gamma)],$$

$$s_{q,d}^C = \operatorname*{argmax}_{s^C} \sum_{s\in S}\sum_{i=1}^{M} I(q^s = q, d_i^s = d)$$
$$[P(S_i^{C,s} = 1, C_i^s = 1|C^s, V^s)\log(s^C) + P(S_i^{C,s} = 0, C_i^s = 1|C^s, V^s)\log(1-s^C)],$$

$$s_{q,d}^E = \operatorname*{argmax}_{s^E} \sum_{s\in S}\sum_{i=1}^{M} I(q^s = q, d_i^s = d)$$
$$[P(S_i^{E,s} = 1, E_i^s = 1, A_i^s = 1, N_i^s = 0|C^s, V^s)\log(s^E)$$
$$+ P(S_i^{E,s} = 0, E_i^s = 1, A_i^s = 1, N_i^s = 0|C^s, V^s)\log(1-s^E)].$$

For the update of the viewport time-related parameters Θ, we use VTCM$_c$ with Weibull distribution for example. In the M-step, we need to update the parameters of Weibull distribution $\Theta = \{\lambda, k\}$ as follows:

$$\lambda_v^{E=0}, k_v^{E=0} = \operatorname*{argmax}_{\lambda,k} \sum_{s\in S}\sum_{i=1}^{M} I(v^s = v)$$
$$P(E_i^s = 0|C^s, V^s)\log\left(f_v^{E=0}(t_i^s)\right),$$

$$\lambda_v^{E=1,C=0,S^E=0}, k_v^{E=1,C=0,S^E=0} = \operatorname*{argmax}_{\lambda,k} \sum_{s\in S}\sum_{i=1}^{M} I(v^s = v)$$
$$P(E_i^s = 1, C_i^s = 0, S_i^{E,s} = 0|C^s, V^s)\log(f_v^{E=1,C=0,S^E=0}(t_i^s)),$$
$$\lambda_v^{E=1,C=1,S^E=0}, k_v^{E=1,C=1,S^E=0} = \operatorname*{argmax}_{\lambda,k} \sum_{s\in S}\sum_{i=1}^{M} I(v^s = v)$$
$$P(E_i^s = 1, C_i^s = 1, S_i^{E,s} = 0|C^s, V^s)\log(f_v^{E=1,C=1,S^E=0}(t_i^s)),$$
$$\lambda_v^{E=1,C=0,S^E=1}, k_v^{E=1,C=0,S^E=1} = \operatorname*{argmax}_{\lambda,k} \sum_{s\in S}\sum_{i=1}^{M} I(v^s = v)$$
$$P(E_i^s = 1, C_i^s = 0, S_i^{E,s} = 1|C^s, V^s)\log(f_v^{E=1,C=0,S^E=1}(t_i^s)),$$

where f is the probability density function of Weibull distribution (Equation (27)).

E-step

Because MCM assumes that the latent variable in last step S_{i-1} may determine E_i and S_i, we need to use the forward-backward algorithm to infer the posterior distributions of the latent variables in each search session s.[7] We define the following variables:

$$f_i(x) = P(S_i = x, C_1, C_2, \ldots, C_i),$$
$$b_i(x) = P(C_{i+1}, \ldots, C_M|S_i = x).$$

These two variables can be computed recursively:

$$f_{i+1}(x) = \sum_{x'\in\{0,1\}} f_i(x')P(S_{i+1} = x, C_{i+1}|S_i = x'),$$
$$b_{i-1}(x) = \sum_{x'\in\{0,1\}} b_i(x')P(S_i = x', C_i|S_{i-1} = x).$$

With $f_i(x)$ and $b_i(x)$, we can compute the posterior distributions needed in the E-step. For example, the posterior distributions needed in the update of $s_{q,d}^E$ can be calculated as follows:

$$\begin{aligned}
&P(S_i^E = 1, E_i = 1, A_i = 1, N_i = 0|C_1, C_2, \ldots, C_M)\\
&= \frac{f_{i-1}(0)b_i(1)P(S_i^E = 1, E_i = 1, A_i = 1, N_i = 0, C_i|S_{i-1} = 0)}{\sum_{x\in 0,1} f_i(x)b_i(x)}\\
&= \frac{f_{i-1}(0)b_i(1)}{\sum_{x\in 0,1} f_i(x)b_i(x)} I(C_i = 0)\gamma_{i,d}\alpha_{q,d_i}(1-\beta_{v_i})s_{q,d_i}^E,
\end{aligned}$$

$$\begin{aligned}
&P(S_i^E = 0, E_i = 1, A_i = 1, N_i = 0|C_1, C_2, \ldots, C_M)\\
&= \frac{f_{i-1}(0)b_i(0)P(S_i^E = 0, E_i = 1, A_i = 1, N_i = 0, C_i|S_{i-1} = 0)}{\sum_{x\in 0,1} f_i(x)b_i(x)}\\
&= \frac{f_{i-1}(0)b_i(0)}{\sum_{x\in 0,1} f_i(x)b_i(x)} I(C_i = 0)\gamma_{i,d}\alpha_{q,d_i}(1-\beta_{v_i})\left(1-s_{q,d_i}^E\right).
\end{aligned}$$

[7] We omit the superscript s here for convenience.

The posterior distributions needed in the update of $s^C_{q,d}$ are as follows:

$$\begin{aligned}
&P(S^C_i = 1, E_i = 1, A_i = 1, N_i = 1 | C_1, C_2, \ldots, C_M) \\
&\quad = \frac{f_{i-1}(0) b_i(1) P(S^C_i = 1, E_i = 1, A_i = 1, N_i = 1, C_i | S_{i-1} = 0)}{\sum_{x \in 0,1} f_i(x) b_i(x)} \\
&\quad = \frac{f_{i-1}(0) b_i(1)}{\sum_{x \in 0,1} f_i(x) b_i(x)} I(C_i = 1) \gamma_{i,d} \alpha_{q,d_i} \beta_{v_i} s^C_{q,d_i},
\end{aligned}$$

$$\begin{aligned}
&P(S^C_i = 0, E_i = 1, A_i = 1, N_i = 1 | C_1, C_2, \ldots, C_M) \\
&\quad = \frac{f_{i-1}(0) b_i(0) P(S^C_i = 0, E_i = 1, A_i = 1, N_i = 1, C_i | S_{i-1} = 0)}{\sum_{x \in 0,1} f_i(x) b_i(x)} \\
&\quad = \frac{f_{i-1}(0) b_i(0)}{\sum_{x \in 0,1} f_i(x) b_i(x)} I(C_i = 1) \gamma_{i,d} \alpha_{q,d_i} \beta_{v_i} \left(1 - s^C_{q,d_i}\right).
\end{aligned}$$

In VTCM, the variables $f_i(x)$ and $b_i(x)$ can be defined as follows:

$$\begin{aligned}
f_i(x) &= P(S_i = x, C_1, \ldots, C_i, V_1, \ldots, V_i), \\
b_i(x) &= P(C_{i+1}, \ldots, C_M, V_{i+1}, \ldots, V_M | S_i = x).
\end{aligned}$$

Thus, they can also be computed recursively:

$$\begin{aligned}
f_{i+1}(x) &= \sum_{x' \in \{0,1\}} f_i(x') P(S_{i+1} = x, C_{i+1}, V_{i+1} | S_i = x'), \\
b_{i-1}(x) &= \sum_{x' \in \{0,1\}} b_i(x') P(S_i = x', C_i, V_i | S_{i-1} = x).
\end{aligned}$$

Then the posterior distributions for the updates of $s^E_{q,d}$ and $s^C_{q,d}$ in $VTCM_e$ can be calculated as follows:

$$\begin{aligned}
&P(S^E_i = 1, E_i = 1, A_i = 1, N_i = 0 | C_1, \ldots, C_M, V_1, \ldots, V_M) \\
&\quad = \frac{f_{i-1}(0) b_i(1) P(S^E_i = 1, E_i = 1, A_i = 1, N_i = 0, C_i, V_i | S_{i-1} = 0)}{\sum_{x \in 0,1} f_i(x) b_i(x)} \\
&\quad = \frac{f_{i-1}(0) b_i(1)}{\sum_{x \in 0,1} f_i(x) b_i(x)} I(C_i = 0) \gamma_{i,d} \alpha_{q,d_i} (1 - \beta_{v_i}) s^E_{q,d_i} f^{E=1}_{v_i}(V_i = t_i),
\end{aligned}$$

$$\begin{aligned}
&P(S^E_i = 0, E_i = 1, A_i = 1, N_i = 0 | C_1, \ldots, C_M, V_1, \ldots, V_M) \\
&\quad = \frac{f_{i-1}(0) b_i(0) P(S^E_i = 0, E_i = 1, A_i = 1, N_i = 0, C_i, V_i | S_{i-1} = 0)}{\sum_{x \in 0,1} f_i(x) b_i(x)} \\
&\quad = \frac{f_{i-1}(0) b_i(0)}{\sum_{x \in 0,1} f_i(x) b_i(x)} I(C_i = 0) \gamma_{i,d} \alpha_{q,d_i} (1 - \beta_{v_i}) \left(1 - s^E_{q,d_i}\right) f^{E=1}_{v_i}(V_i = t_i),
\end{aligned}$$

$$
\begin{aligned}
&P(S_i^C = 1, E_i = 1, A_i = 1, N_i = 1 | C_1, \ldots, C_M, V_1, \ldots, V_M) \\
&\quad = \frac{f_{i-1}(0) b_i(1) P(S_i^C = 1, E_i = 1, A_i = 1, N_i = 1, C_i, V_i | S_{i-1} = 0)}{\sum_{x \in 0,1} f_i(x) b_i(x)} \\
&\quad = \frac{f_{i-1}(0) b_i(1)}{\sum_{x \in 0,1} f_i(x) b_i(x)} I(C_i = 1) \gamma_{i,d} \alpha_{q,d_i} \beta_{v_i} s^C_{q,d_i} f^{E=1}_{v_i}(V_i = t_i),
\end{aligned}
$$

$$
\begin{aligned}
&P(S_i^C = 0, E_i = 1, A_i = 1, N_i = 1 | C_1, \ldots, C_M, V_1, \ldots, V_M) \\
&\quad = \frac{f_{i-1}(0) b_i(0) P(S_i^C = 0, E_i = 1, A_i = 1, N_i = 1, C_i, V_i | S_{i-1} = 0)}{\sum_{x \in 0,1} f_i(x) b_i(x)} \\
&\quad = \frac{f_{i-1}(0) b_i(0)}{\sum_{x \in 0,1} f_i(x) b_i(x)} I(C_i = 1) \gamma_{i,d} \alpha_{q,d_i} \beta_{v_i} \left(1 - s^C_{q,d_i}\right) f^{E=1}_{v_i}(V_i = t_i).
\end{aligned}
$$

In VTCM$_c$, the four posterior distributions are as follows:

$$
\begin{aligned}
&P(S_i^E = 1, E_i = 1, A_i = 1, N_i = 0 | C_1, \ldots, C_M, V_1, \ldots, V_M) \\
&\quad = \frac{f_{i-1}(0) b_i(1) P(S_i^E = 1, E_i = 1, A_i = 1, N_i = 0, C_i, V_i | S_{i-1} = 0)}{\sum_{x \in 0,1} f_i(x) b_i(x)} \\
&\quad = \frac{f_{i-1}(0) b_i(1)}{\sum_{x \in 0,1} f_i(x) b_i(x)} I(C_i = 0) \gamma_{i,d} \alpha_{q,d_i} (1 - \beta_{v_i}) s^E_{q,d_i} f^{E=1, C=0, S^E=1}_{v_i}(V_i = t_i),
\end{aligned}
$$

$$
\begin{aligned}
&P(S_i^E = 0, E_i = 1, A_i = 1, N_i = 0 | C_1, \ldots, C_M, V_1, \ldots, V_M) \\
&\quad = \frac{f_{i-1}(0) b_i(0) P(S_i^E = 0, E_i = 1, A_i = 1, N_i = 0, C_i, V_i | S_{i-1} = 0)}{\sum_{x \in 0,1} f_i(x) b_i(x)} \\
&\quad = \frac{f_{i-1}(0) b_i(0)}{\sum_{x \in 0,1} f_i(x) b_i(x)} I(C_i = 0) \gamma_{i,d} \alpha_{q,d_i} (1 - \beta_{v_i}) \left(1 - s^E_{q,d_i}\right) f^{E=1, C=0, S^E=0}_{v_i}(V_i = t_i),
\end{aligned}
$$

$$
\begin{aligned}
&P(S_i^C = 1, E_i = 1, A_i = 1, N_i = 1 | C_1, \ldots, C_M, V_1, \ldots, V_M) \\
&\quad = \frac{f_{i-1}(0) b_i(1) P(S_i^C = 1, E_i = 1, A_i = 1, N_i = 1, C_i, V_i | S_{i-1} = 0)}{\sum_{x \in 0,1} f_i(x) b_i(x)} \\
&\quad = \frac{f_{i-1}(0) b_i(1)}{\sum_{x \in 0,1} f_i(x) b_i(x)} I(C_i = 1) \gamma_{i,d} \alpha_{q,d_i} \beta_{v_i} s^C_{q,d_i} f^{E=1, C=1, S^C=1}_{v_i}(V_i = t_i),
\end{aligned}
$$

$$
\begin{aligned}
&P(S_i^C = 0, E_i = 1, A_i = 1, N_i = 1 | C_1, \ldots, C_M, V_1, \ldots, V_M) \\
&\quad = \frac{f_{i-1}(0) b_i(0) P(S_i^C = 0, E_i = 1, A_i = 1, N_i = 1, C_i, V_i | S_{i-1} = 0)}{\sum_{x \in 0,1} f_i(x) b_i(x)} \\
&\quad = \frac{f_{i-1}(0) b_i(0)}{\sum_{x \in 0,1} f_i(x) b_i(x)} I(C_i = 1) \gamma_{i,d} \alpha_{q,d_i} \beta_{v_i} \left(1 - s^C_{q,d_i}\right) f^{E=1, C=1, S^C=0}_{v_i}(V_i = t_i).
\end{aligned}
$$

ACKNOWLEDGMENTS

We thank *Sogou.com* for the anonymized mobile search log used in this work and our anonymous reviewers for their helpful comments and valuable suggestions.

REFERENCES

[1] Alexey Borisov, Ilya Markov, Maarten de Rijke, and Pavel Serdyukov. 2016. A neural click model for web search. In *Proceedings of the WWW'16*. International World Wide Web Conferences Steering Committee, 531–541.

[2] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected reciprocal rank for graded relevance. In *Proceedings of the CIKM'09*. ACM, 621–630.

[3] Olivier Chapelle and Ya Zhang. 2009. A dynamic Bayesian network click model for web search ranking. In *Proceedings of the WWW'09*. ACM, 1–10.

[4] Danqi Chen, Weizhu Chen, Haixun Wang, Zheng Chen, and Qiang Yang. 2012. Beyond ten blue links: Enabling user click modeling in federated web search. In *Proceedings of the WSDM'12*. ACM, 463–472.

[5] Aleksandr Chuklin and Maarten de Rijke. 2016. Incorporating clicks, attention, and satisfaction into a search engine result page evaluation model. In *Proceedings of the CIKM'16*. ACM, 175–184.

[6] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. Click models for web search. *Synth. Lect. Inform. Conc., Retr., Serv.* 7, 3 (2015), 1–115.

[7] Aleksandr Chuklin, Pavel Serdyukov, and Maarten De Rijke. 2013. Using intent information to model user behavior in diversified search. In *Proceedings of the ECIR'13*. Springer, 1–13.

[8] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *Proceedings of the WSDM'08*. ACM, 87–94.

[9] Georges E. Dupret and Benjamin Piwowarski. 2008. A user browsing model to predict search engine click data from past observations. In *Proceedings of the SIGIR'08*. ACM, 331–338.

[10] Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychol. Bull.* 76, 5 (1971), 378.

[11] Laura A. Granka, Thorsten Joachims, and Geri Gay. 2004. Eye-tracking analysis of user behavior in WWW search. In *Proceedings of the SIGIR'04*. ACM, 478–479.

[12] Fan Guo, Chao Liu, and Yi Min Wang. 2009. Efficient multiple-click models in web search. In *Proceedings of the WSDM'09*. ACM, 124–131.

[13] Qi Guo, Haojian Jin, Dmitry Lagun, Shuai Yuan, and Eugene Agichtein. 2013. Mining touch interaction data on mobile devices to predict web search result relevance. In *Proceedings of the SIGIR'13*. ACM, 153–162.

[14] Morgan Harvey and Matthew Pointon. 2017. Searching on the go: The effects of fragmented attention on mobile Web search tasks. In *Proceedings of the SIGIR'17*. ACM, 155–164.

[15] Jeff Huang, Ryen W. White, Georg Buscher, and Kuansan Wang. 2012. Improving searcher models using mouse cursor activity. In *Proceedings of the SIGIR'12*. ACM, 195–204.

[16] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inform. Syst.* 20, 4 (2002), 422–446.

[17] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately interpreting click-through data as implicit feedback. In *Proceedings of the SIGIR'05*. ACM, 154–161.

[18] Maryam Kamvar and Shumeet Baluja. 2006. A large scale study of wireless search behavior: Google mobile search. In *Proceedings of the SIGCHI'06*. ACM, 701–709.

[19] Maryam Kamvar, Melanie Kellar, Rajan Patel, and Ya Xu. 2009. Computers and iphones and mobile phones, oh my!: A logs-based comparison of search users on different devices. In *Proceedings of the WWW'09*. ACM, 801–810.

[20] Jaewon Kim, Paul Thomas, Ramesh Sankaranarayana, Tom Gedeon, and Hwan-Jin Yoon. 2015. Eye-tracking analysis of user behavior and performance in web search on large and small screens. *J. Assoc. Inform. Sci. Technol.* 66, 3 (2015), 526–544.

[21] Klaus Krippendorff. 1970. Estimating the reliability, systematic error and random error of interval data. *Educat. Psychol. Meas.* 30, 1 (1970), 61–70.

[22] Dmitry Lagun, Chih-Hung Hsieh, Dale Webster, and Vidhya Navalpakkam. 2014. Towards better measurement of attention and satisfaction in mobile search. In *Proceedings of the SIGIR'14*. ACM, 113–122.

[23] Dmitry Lagun, Donal McMahon, and Vidhya Navalpakkam. 2016. Understanding mobile searcher attention with rich ad formats. In *Proceedings of the CIKM'16*. ACM, 599–608.

[24] J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics* 33, 1 (1977), 159–174.

[25] Yiqun Liu, Xiaohui Xie, Chao Wang, Jian-Yun Nie, Min Zhang, and Shaoping Ma. 2016. Time-aware click model. *ACM Trans. Inform. Syst.* 35, 3 (Dec. 2016), 16:1–16:24.

[26] Zeyang Liu, Yiqun Liu, Ke Zhou, Min Zhang, and Shaoping Ma. 2015. Influence of vertical result in web search examination. In *Proceedings of the SIGIR'15*. ACM, 193–202.

[27] Zeyang Liu, Jiaxin Mao, Chao Wang, Qingyao Ai, Yiqun Liu, and Jian-Yun Nie. 2017. Enhancing click models with mouse movement information. *Inform. Retr. J.* 20, 1 (2017), 53–80.

[28] Cheng Luo, Yiqun Liu, Tetsuya Sakai, Fan Zhang, Min Zhang, and Shaoping Ma. 2017. Evaluating mobile search with height-biased gain. In *Proceedings of the SIGIR'17*. ACM, 435–444.

[29] Jiaxin Mao, Yiqun Liu, Noriko Kando, Cheng Luo, Min Zhang, and Shaoping Ma. 2018. Investigating result usefulness in mobile search. In *Proceedings of the ECIR'18*. Springer, 223–236.
[30] Jiaxin Mao, Cheng Luo, Min Zhang, and Shaoping Ma. 2018. Constructing click models for mobile search. In *Proceedings of the SIGIR'18*. ACM, 775–784.
[31] Vidhya Navalpakkam, LaDawn Jentzsch, Rory Sayres, Sujith Ravi, Amr Ahmed, and Alex Smola. 2013. Measurement and modeling of eye-mouse behavior in the presence of nonlinear page layouts. In *Proceedings of the WWW'13*. ACM, 953–964.
[32] Kevin Ong, Kalervo Järvelin, Mark Sanderson, and Falk Scholer. 2017. Using information scent to understand mobile and desktop Web search behavior. In *Proceedings of the SIGIR'17*. ACM, 295–304.
[33] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: Estimating the click-through rate for new ads. In *Proceedings of the WWW'07*. ACM, 521–530.
[34] Tetsuya Sakai. 2007. Alternatives to bpref. In *Proceedings of the SIGIR'07*. ACM, 71–78.
[35] Yang Song, Hao Ma, Hongning Wang, and Kuansan Wang. 2013. Exploring and exploiting user search behavior on mobile and tablet devices to improve search relevance. In *Proceedings of the WWW'13*. ACM, 1201–1212.
[36] Manisha Verma and Emine Yilmaz. 2016. Characterizing relevance on mobile and desktop. In *Proceedings of the ECIR'16*. Springer, 212–223.
[37] Chao Wang, Yiqun Liu, Meng Wang, Ke Zhou, Jian-yun Nie, and Shaoping Ma. 2015. Incorporating non-sequential behavior into click models. In *Proceedings of the SIGIR'15*. ACM, 283–292.
[38] Chao Wang, Yiqun Liu, Min Zhang, Shaoping Ma, Meihong Zheng, Jing Qian, and Kuo Zhang. 2013. Incorporating vertical results into search click models. In *Proceedings of the SIGIR'13*. ACM, 503–512.
[39] Xiaochuan Wang, Ning Su, Zexue He, Yiqun Liu, and Shaoping Ma. 2018. A large-scale study of mobile search examination behavior. In *Proceedings of the SIGIR'18*. ACM, 1129–1132.
[40] Kyle Williams, Julia Kiseleva, Aidan C. Crook, Imed Zitouni, Ahmed Hassan Awadallah, and Madian Khabsa. 2016. Detecting good abandonment in mobile search. In *Proceedings of the WWW'16*. 495–505.
[41] Kyle Williams, Julia Kiseleva, Aidan C. Crook, Imed Zitouni, Ahmed Hassan Awadallah, and Madian Khabsa. 2016. Is this your final answer?: Evaluating the effect of answers on good abandonment in mobile search. In *Proceedings of the SIGIR'16*. ACM, 889–892.
[42] Wan-Ching Wu, Diane Kelly, and Avneesh Sud. 2014. Using information scent and need for cognition to understand online search behavior. In *Proceedings of the SIGIR'14*. ACM, 557–566.
[43] Jeonghee Yi, Farzin Maghoul, and Jan Pedersen. 2008. Deciphering mobile search patterns: A study of Yahoo! mobile search queries. In *Proceedings of the WWW'08*. ACM, 257–266.

Received October 2018; revised June 2019; accepted July 2019