

# Personalized Click Model through Collaborative Filtering

Si Shen<sup>\*</sup>  
Hong Kong University of  
Science and Technology  
shensi@ust.hk

Weizhu Chen  
Microsoft Research Asia  
wzchen@microsoft.com

Botao Hu  
Institute for Interdisciplinary  
Information Sciences  
Tsinghua University  
botao.a.hu@gmail.com

Qiang Yang  
Hong Kong University of  
Science and Technology  
qyang@cse.ust.hk

## ABSTRACT

Click modeling aims to interpret the users' search click data in order to predict their clicking behavior. Existing models can well characterize the position bias of documents and snippets in relation to users' mainstream click behavior. Yet, current advances depict users' search actions only in a general setting by implicitly assuming that all users act in the same way, regardless of the fact that anyone, motivated with some individual interest, is more likely to click on a link than others. It is in light of this that we put forward a novel *personalized click model* to describe the user-oriented click preferences, which applies and extends matrix / tensor factorization from the view of collaborative filtering to connect users, queries and documents together. Our model serves as a generalized personalization framework that can be incorporated to the previously proposed click models and, in many cases, to their future extensions. Despite the sparsity of search click data, our personalized model demonstrates its advantage over the best click models previously discussed in the Web-search literature, supported by our large-scale experiments on a real dataset. A delightful bonus is the model's ability to gain insights into queries and documents through latent feature vectors, and hence to handle rare and even new query-document pairs much better than previous click models.

## Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Retrieval Models

---

<sup>\*</sup>This work is done during the author's internship at Microsoft Research Asia

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSDM'12, February 8–12, 2012, Seattle, Washington, USA.  
Copyright 2012 ACM 978-1-4503-0747-5/12/02 ...\$10.00.

## General Terms

Algorithms

## Keywords

Click Model, Click Log Analysis, Personalization, Search Engine, User Behavior

## 1. INTRODUCTION

Click-through logs in search engines are a valuable resource for learning user preferences for search or advertisement (ads) results. The analysis of these log data can facilitate a number of search-related applications, such as Web search ranking, ads click-through rate (CTR) prediction and optimization, and user satisfaction estimation. The relevant research can not only enhance user search experience, but also make an impact on search revenue as well.

Click prediction aims at computing the probability that a given document in a search-result list is clicked on after a user enters some query. To accurately predict user clicks, a central question is to resolve the user-perceived relevance between query-document pairs. Having learnt this relevance from massive search click data, a commercial search engine can improve its understanding of search users and therefore polish its search results. Recently, the problem of learning user-perceived document relevance from click-through data has been formalized as a problem of click model learning, and has become an attractive research topic. Many studies have been attempted to mine the general user preferences from click-through logs to improve the overall relevance of search results [14, 13, 2].

Since click data are informative and can be easily collected, they offer a promising approach to optimize search engine performance with low cost. However, a well-known challenge of using such data is the position bias, whereby a document at a higher position on a Web page tends to attract more user clicks even when in the case of low document relevance. This bias was firstly mentioned by [9] in their eye-tracking experiment, which pointed out that the frequently used CTR may not be appropriate as an estimator of document relevance. Prior work [19] in sponsored advertisement search attempted to exploit the position bias by imposing a multiplicative factor on documents at lower positions to infer their true relevance scores. This idea was further explored in organic searches [5], and was formalized as the

so-called *Examination Hypothesis*, a fundamental assumption for many following important models. For example, these models include the User Browsing Model (UBM) [8], Dynamic Bayesian Network (DBN) model [3], Click Chain Model (CCM) [10], Bayesian Browsing Model (BBM) [17], General Click Model (GCM) [24] and the Session Utility Model (SUM) [7].

In spite of their merits, these previous developments in click modeling are limited by their assumption that the query-document relevance model is identical for all search engine users. However, we observe that there are a large number of similar queries submitted by different users that resulted in very different information needs and search preferences [12]. In response, in this paper, we argue that a single global query-document relevance is not sufficient to truly reflect the interest of every individual user. Therefore, we must develop a new method for building personalized relevance models to fully capture diversified user interests.

To further illustrate our motivation, take a query “opera” as an example. This query may come from an engineer, who expects results that present the web browser Opera at `opera.com`. In contrast, a music lover who submits the same query may look for some vocal concert performance. In this situation, search results like `sfopera.com` and other opera houses are more likely to be clicked. This example indicates that the attractiveness of a search result is not only influenced by its relevance but also determined by the user’s intrinsic search need for the query. Furthermore, notice that the engineer might have searched for technological software for a couple of times previously, and that the music lover could be a regular visitor to music Web sites. In this paper, we show that those activities that are embedded in the search history logs can be exploited to imply users’ personal search preferences. Thus, we can analyze this history data to provide personalized search results to meet user’s interests.

An intuitive approach to solve the personalization problem is to utilize the user’s historical search data to learn personalized document relevance models. However, a major challenge is that user click data for a query-document pair are often grossly insufficient and sparse. Existing click models consider query-document as an integrated pair, but do not have the ability to handle such sparse data. This limitation leads us to a different approach, in which we capture user preferences from click logs *collaboratively*. In particular, we propose a collaborative-filtering based click-model framework. In this framework, we factorize the user-query-document relations as latent vectors, which reflect the diversity of user’s intrinsic preferences and the inter- and intra-relationships of queries and documents. In the process, we take the personalized click model under a general framework user modeling, for which we pay attention to personalize document relevance and leave position biases intact. Note that the two parts of Examination Hypothesis remain relatively independent, so that not only different depictions of position bias from previous click models can be replaced in the model, but various collaborative filtering models may substitute one another as well.

The contribution of this paper is two-fold.

- We address the issue of personalization for click models by introducing collaborative filtering to the literature of web search. Our model is able to handle low frequency and even new query-document combinations that preceding click models cannot manage,

for we introduce matrix factorization, and its extension tensor factorization, as a mature and highly scalable model. Queries and documents are then implicitly connected through their latent feature vectors and further through users. As a result, new query-document pairs that are not present in training data may still be inferred, and informational queries with complex click logs can be predicted more precisely. We believe that this is the first time for this to be done.

- We conduct extensive experiments on large-scale real-world click log data to evaluate our model and algorithm. Our results show that the personalization of click model is a highly promising direction for click-modeling research.

In the rest of this paper, we first introduce some of the existing click models and related theories in Section 2. We then explain our factorization models, as well as the inference method in details in Section 3. We perform experiments on the personalized click models based on real dataset and present our results and insights in Section 4 and Section 5, respectively. We share our thoughts of possible future work and conclude the paper in Section 6.

## 2. BACKGROUND

In this section, we first introduce the basic Position Model and then explore its extensions to various existing click models.

Before delving into the detailed introduction of click models, we describe some relevant concepts when a user submits a search query and the notations used throughout the paper. A user  $u$  first makes an inquiry by sending query  $q$ . After fetching Search Engine Result Page (SERP), one scrutinizes the provided documents, or referred as urls in some work [5], in a particular order and decides to either click or skip a document  $d$ . The widely accepted assumption is that the decision is made according to the user-perceived relevance between document and query. In general, the prediction of user click behavior is modeled in a probabilistic setting. It should be noted that SERP returns not only the organic search results mentioned above, but also other links like ads and related queries that a user might perform a click [4]. To simplify the problem, such links are out of the scope of this paper.

We introduce notations used in the paper. Each observation of click contains a vector  $\{C, u, q, d, i\}$ , where  $C = 1$  means a click and  $i$  is the document position. The data set  $\mathbf{S}$  in query logs are partitioned into  $\mathbf{S}^\bullet$ , the subset that contains all the clicked items where  $C = 1$ , and its complement  $\mathbf{S}^\circ$ , the subset of observations that are not clicked. Within our paper, we follow the notations in [8] where a dot denotes a click and a hallow circle a skip. The index of  $\mathbf{S}$  denotes the requirement that the observations need to meet to be in the set. For example,  $\mathbf{S}_{q,d,i}^\bullet$  refers to all the observations such that document  $d$  at position  $i$  for the given query  $q$  is clicked. Similarly,  $\mathbf{S}_i^\circ$  represents all the observations such that a document at position  $i$  is neglected. Additionally, we designate  $S^\bullet$  and  $S^\circ$  the cardinalities of the corresponding sets.

### 2.1 Position Model

The Position Model [19] is a broadly accepted basic click model. It incorporates position bias by assuming that a user

makes the click decision only after examining the document’s title and snippet. Whether the user examines a document item or not depends on the position of the document. This is referred as the *Examination Hypothesis*.

If we let  $C = 1$  denote the event of click and value 0 the absence of the event, then the probability to click a document  $d$  at position  $i$  given query  $q$  can be expressed as  $P(C_i = 1|q, d)$ .

Now let  $E = 1$  denote the event of examination and value 0 disregard for the document, and the probability of click under Examination Hypothesis can be formulated as

$$\begin{aligned} P(C_i = 1|q, d) &= \sum_{E_i} P(C_i = 1|E_i, q, d)P(E_i) \\ &= \underbrace{P(C_i = 1|E_i = 1, q, d)}_{\text{document relevance}} \underbrace{P(E_i = 1)}_{\text{position bias}} \end{aligned}$$

Define

$$\alpha_{qd} = P(C_i = 1|E_i = 1, q, d) \text{ and } \beta_i = P(E_i = 1) ,$$

then

$$\begin{aligned} P(C_i = 1|q, d) &= \alpha_{qd}\beta_i \\ P(C_i = 0|q, d) &= \sum_E P(C = 0|E, q, d)P(E_i) \\ &= 1 - \alpha_{qd}\beta_i \end{aligned}$$

From the above, the click event follows a simple Bernoulli distribution. When the parameter set  $\beta$  are set as ones, the model degrades to a baseline model that a click depends solely on the user-perceived relevance of the document to a given query. The joint likelihood under our notations is then

$$\mathcal{L}(\alpha, \beta|\mathbf{S}) = \prod_{q,d,i} (\alpha_{qd}\beta_i)^{S_{qd,i}^\bullet} (1 - \alpha_{qd}\beta_i)^{S_{qd,i}^\circ}$$

In this model, the hidden variable is the event of examination  $E$ . One can apply Expectation Maximization (EM) algorithm to estimate  $\alpha$  and  $\beta$  in an alternation fashion [6]. This method can also avoid the possible situations when  $\alpha > 1$ .

## 2.2 Cascade Model

The pure Position Model copes only with different positions that are independent of each other. Many extensions were subsequently proposed to take the whole-session interactions into consideration. One session here is defined as all the search results returned for one requested query. The number of relevant documents within a session is normally set as 10. More hypotheses were raised to interpret hidden information in a session.

A prominent hypothesis is that users tend to examine documents one by one from top of the page. A document can be examined only if its upper neighbor is examined. An additional condition that a user stops the session when a document is clicked gives Cascade Model [15]. Under these requirements, a click on the document indicates that: (1) the user has examined all the documents above and regarded them as irrelevant; (2) all the documents below will not be examined and the session is terminated. Again, it is quite clear that whether a user submits a click depends on the user-perceived document relevance just as in Position Model.

Following prior notations, Cascade Model can be formalized.

$$\begin{aligned} P(E_i = 1) &= 1 \\ P(E_{i+1} = 1|E_i = 0) &= 0 \\ P(C_i = 1|E_i = 1) &= \alpha_{qd_i} \\ P(E_{i+1} = 1|E_i = 1, C_i) &= 1 - C_i \end{aligned}$$

Then the probability of click can be computed as

$$P(C_i = 1) = \alpha_{qd_i} \prod_{j=1}^{i-1} (1 - \alpha_{qd_j})$$

While the Position Model focuses on depicting position bias as precisely as possible, the Cascade Hypothesis was proposed to view user search behavior as an action chain. One drawback of the Cascade Model is that the assumption is so strict that sessions with more than one click are to be discarded. This restriction gives rise to some subsequent extensions, which we review below.

## 2.3 User Browsing Model

The hypothesis raised by the User Browsing Model (UBM) [8] aims at arriving at a better estimation of position bias by relaxing it from the limitation of constant multipliers. It also overcomes the constraint of Cascade Model that only one-click sessions can be handled. The model says that position bias depends not only on current position  $i$ , but also on the distance to the position of the latest clicked document  $r$ .  $\beta_i$  is hence extended to  $\beta_{ir}$ .

We set  $r = 0$  if there is no preceding clicks. If we let  $\alpha_{qd_i}$  denote the user-perceived relevance of document  $d$  at position  $i$  for a given query  $q$ , then UBM can be formally illustrated as follows.

$$\begin{aligned} P(C_i = 1|E_i = 1) &= \alpha_{qd_i} \\ P(C_i = 1|E_i = 0) &= 0 \\ P(E_i = 1|C_{i-r} = 1, C_{i-r+1} = \dots = C_{i-1} = 0) &= \beta_{ir} \\ P(E_i = 1|C_1 = \dots = C_{i-1} = 0) &= \beta_{i0} \end{aligned}$$

The joint likelihood bears a form similar to Position Model.

$$\mathcal{L}(\alpha, \beta|\mathbf{S}) = \prod_{i,r} \prod_{q,d} (\alpha_{qd}\beta_{ir})^{S_{qd,ir}^\bullet} (1 - \alpha_{qd}\beta_{ir})^{S_{qd,ir}^\circ}$$

Similarly, we can also use Expectation Maximization (EM) algorithm to complete model inference [8].

## 3. PERSONALIZED CLICK MODEL

In this section, we present our main approach for personalized click models. We first analyze the drawbacks of current click models and then introduce collaborative filtering to the click-through-rate prediction problem. We then consider three different matrix or tensor factorization set-ups. The first one is a matrix factorization model on queries and documents only to compare with preceding click models to show the capability of our decomposition method. The model has self-explanatory probability meanings. The second one is a direct incorporation of probabilistic tensor factorization on users, queries and documents to illustrate the framework of personalized click model. The last one is a hybrid factorization model that emphasizes on queries and documents interaction and characterizes user preference through user’s deviation from average. We give reasons for why the hybrid

model is superior to the other two, and the inference is given after the models.

### 3.1 Limitation of Previous Works

When looking back at the preceding click models, we notice two main drawbacks. It was these two limitations that inspires our new models. Firstly, while existing models put quite some efforts to depict position bias, the complexity of document relevance has been largely neglected. Individual characteristics of queries and documents were not thoroughly considered. A query and a document has been simply treated as an integrated pair. Hence, a new query-document combination in testing data cannot be effectively handled.

Another problem is that previous click models regard document relevance as a global parameter and overlook any possible disparity due to user personal behavior. However, user  $u$  may have personal interest in or preference to a particular url entry  $d$  of query  $q$ , and consequently, CTR of this query-document pair by  $u$  can be relatively higher than average.

To relax the limitation of query-document pair perspective and to incorporate personalization to click models, our novel approach adopts some ideas from recommender systems [20, 16, 18].

The general setting of a recommender system consists of two main components, users and items. A user  $u$  may have rated an item  $i$ , with score  $r_{ui}$ . The score associated with that user-item pair can be a numerical value, or a 0/1 indicator. Yet, there are also many unrated user-item pairs. The goal of a recommender system is to use existing rated values  $\{r_{ui}\}$  to predict scores for requisite user-item pairs with missing values.

Collaborative filtering is an important technique that explores user similarity to compute new ratings for products in recommender systems. Among many of the models of collaborative filtering, we use matrix factorization, and its tensor extension, in our work for two reasons: (1) matrix factorization characterizes latent factors of both items and users implied from feedback log; (2) matrix factorization is scalable and flexible with high accuracy. By using matrix factorization, we can not only handle new query-document pairs from latent factors and implicit connections, but effectively and efficiently solve the problem of personalization as well.

### 3.2 Matrix Factorization Click Model

As introduced in Section 3.1, one limitation of previous work neglects interactions between queries and documents, which are simply modeled as pairs. We now propose a matrix factorization click model (MFCM) to focus on queries and documents interactions through their latent feature vectors. Suppose that a query  $q$  is submitted in a session and  $N$  documents are fetched, the  $i$ -th document being  $d_i$ . If there are in total  $M_q$  queries and  $M_d$  documents, then we let  $Q \subset \mathbb{R}^{F \times M_q}$ , where the set consists of all possible real-valued matrices with size  $F \times M_q$ , and  $D \subset \mathbb{R}^{F \times M_d}$  represents the latent factors of queries and documents, respectively. In this equation,  $F$  is the number of factors. Let  $\mathcal{N}(\mu, \sigma^2)$  be the probability density function of Gaussian distribution with mean  $\mu$  and variance  $\sigma^2$ .

We adopt a probabilistic linear model with Gaussian observation noise, and place zero-mean spherical Gaussian priors on queries and documents latent factors. MFCM is con-

structed as

$$P(C_i = 1) = P(C_i = 1|E_i = 1)P(E_i = 1)$$

$$P(C_i = 1|E_i = 1) = \alpha_{qd_i}$$

$$P(\alpha_{qd_i}|Q_q, D_d, \sigma) \sim \mathcal{N}((Q_q \circ D_{d_i}), \sigma^2)$$

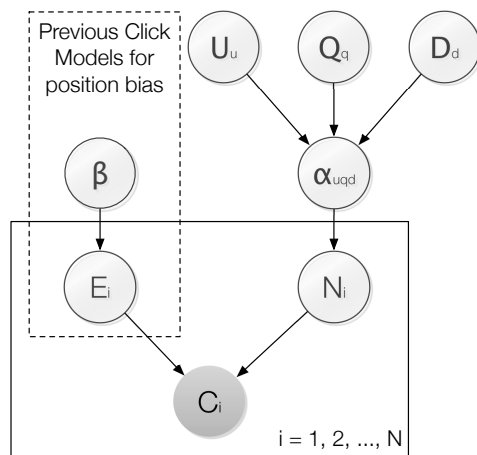
$$P(Q_q|\sigma_Q) \sim \mathcal{N}(0, \sigma_Q^2 I)$$

$$P(D_d|\sigma_D) \sim \mathcal{N}(0, \sigma_D^2 I)$$

Although MFCM collaboratively solves the first limitation of previous works, it does not introduce personalization to click models. Hence, a tensor factorization click model with users as another dimension is presented next.

### 3.3 Tensor Factorization Click Model

To include personalization in our click models, we introduce a Personalized Click Model (PCM) in this section to extend MFCM to the user domain. Suppose that there are  $M_u$  users. Let  $U \subset \mathbb{R}^{F \times M_u}$  denote the latent factors of user domain. The event of user being personally interested in the  $i$ -th document is indicated by  $N_i$ . Let  $\alpha_{uqd_i}$  denote the probability of event  $N_i$ .



**Figure 1: Graphic representation of Personalized Click Model**

We impose again zero-mean spherical Gaussian priors on user latent factors, and extend matrix factorization to tensor decomposition, which naturally produces personalized click predictions. Our PCM can be characterized by the following group of equations.

$$C_i = 1 \Leftrightarrow E_i = 1, N_i = 1 \tag{1}$$

$$P(N_i = 1) = \alpha_{uqd_i} \tag{2}$$

$$P(\alpha_{uqd_i}|U_u, Q_q, D_d, \sigma) \sim \mathcal{N}((U_u \circ Q_q \circ D_{d_i}), \sigma^2) \tag{3}$$

$$P(U_u|\sigma_U) \sim \mathcal{N}(0, \sigma_U^2 I) \tag{4}$$

$$P(Q_q|\sigma_Q) \sim \mathcal{N}(0, \sigma_Q^2 I) \tag{5}$$

$$P(D_d|\sigma_D) \sim \mathcal{N}(0, \sigma_D^2 I) \tag{6}$$

Note that  $U_u \circ Q_q \circ D_{d_i} = \sum_{f=1}^F U_{fu} Q_{fq} D_{fd_i}$  as a traditional canonical tensor factorization.

Figure 1 gives a graphic representation of PCM, showing the connections of parameters. Gaussian priors are omitted from the graph model. As Examination Hypothesis suggested, the event of click at position  $i$  is dependent on the event of user examination and an individual document interest from user  $u$ . The examination of a document can still

follow various assumptions of previous click models, while the personal document interest is modeled as a probability based on the latent factors of users, queries and documents themselves.

PCM is promising intuitively by considering the implicit interactions among users, queries and documents. However, for a number of navigational queries, personal differentiation is relatively insignificant. A fully personalized click model may occasionally suffer from over-fit problem.

### 3.4 Hybrid Personalized Model

In light of the capabilities and limitations of both MFCM and PCM, in this section, we put forward a hybrid personalized click model (HPCM). Instead of performing a direct canonical tensor factorization, we place our emphasis on the interactions between queries and documents, which have been believed to be the dominant part for relevance determination. Then to solve the problem of personalization, only the residuals are factorized using user latent factors to describe personal deviations from the global query-document factor model. That is, HPCM is a combination of PCM and MFCM. The conditional distribution of  $\alpha_{uqd_i}$  can be modeled as

$$P(\alpha_{uqd_i} | \tilde{Q}_q, \tilde{D}_d, U_u, Q_q, D_d, \sigma) \\ \sim \mathcal{N} \left( \underbrace{\tilde{Q}_q \circ \tilde{D}_d}_{\text{query-doc bias}} + \underbrace{U_u \circ Q_q \circ D_d}_{\text{user diversity}}, \sigma^2 \right)$$

Gaussian priors are

$$P(\tilde{Q}_q | \sigma_{\tilde{Q}}) \sim \mathcal{N}(0, \sigma_{\tilde{Q}}^2 I) \\ P(\tilde{D}_d | \sigma_{\tilde{D}}) \sim \mathcal{N}(0, \sigma_{\tilde{D}}^2 I) \\ P(U_u | \sigma_U) \sim \mathcal{N}(0, \sigma_U^2 I) \\ P(Q_q | \sigma_Q) \sim \mathcal{N}(0, \sigma_Q^2 I) \\ P(D_d | \sigma_D) \sim \mathcal{N}(0, \sigma_D^2 I)$$

The interactions between queries and documents can be viewed as a relevance bias, while the user-query-document relationship may be viewed as user preference variations. Personalization is implemented after a global inference of query-document relevance. This combined factor model can therefore outperform MFCM and PCM, illustrated in Section 4. The graphic model of HPCM can be found in Figure 2.

### 3.5 Inference

Due to space limitation, we present inference process using Position Model describing position bias, that is,  $\beta_i$ . Inference for other position bias models are very similar and can be derived accordingly. Taking PCM as an example, the parameters to be inferred are

$$\Theta = \{\alpha, \beta, U, Q, D\}$$

Define

$$R_{uqd} \equiv \exp \left\{ -\frac{(\alpha_{uqd} - U_u \circ Q_q \circ D_d)^2}{2\sigma^2} - \frac{U_u^T U_u}{2\sigma_U^2} \right. \\ \left. - \frac{Q_q^T Q_q}{2\sigma_Q^2} - \frac{D_d^T D_d}{2\sigma_D^2} \right\}$$

The conditional joint probability of observed data is then

$$P(\mathbf{S} | \Theta) \propto \prod_{u,q,d,i} (\alpha_{uqd} \beta_i)^{C_i} (1 - \alpha_{uqd} \beta_i)^{1-C_i} R_{uqd}$$

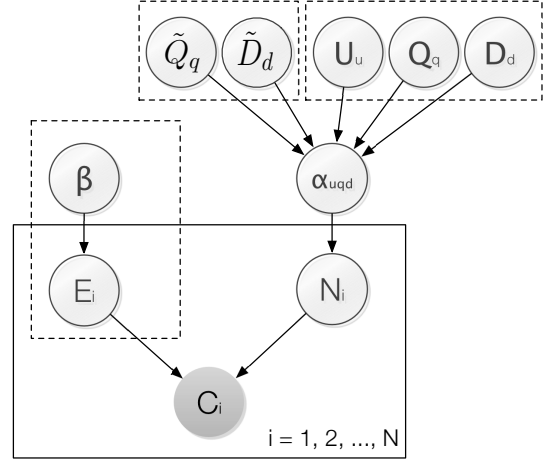


Figure 2: Graphic representation of Hybrid Personalized Click Model

The log-likelihood can be computed as

$$\log \mathcal{L}(\Theta | \mathbf{S}, \sigma^2, \sigma_U^2, \sigma_Q^2, \sigma_D^2) \\ = \sum_{u,q,d,i} S_{uqd,i}^* (\alpha_{uqd} \beta_i) + \sum_{u,q,d,i} S_{uqd,i}^o (1 - \alpha_{uqd} \beta_i) \\ - \frac{1}{2\sigma^2} \sum_{u,q,d} S_{uqd} (\alpha_{uqd} - U_u \circ Q_q \circ D_d)^2 \\ - \frac{1}{2\sigma_U^2} \sum_u S_u U_u^T U_u - \frac{1}{2\sigma_Q^2} \sum_q S_q Q_q^T Q_q \\ - \frac{1}{2\sigma_D^2} \sum_d S_d D_d^T D_d + \text{constant}$$

EM algorithm is utilized, with the event of examination as hidden variable, to estimate the parameters maximizing the above log likelihood. A more detailed inference procedure can be found in Appendix.

Three observations can be made from the inference process.

- Position bias  $\beta$  has a rather independent inference process, which implies that previous click models portraying this bias can be substituted into the framework.
- The inference of latent vectors can also be implemented independently to some extent, allowing various factor models to be included. In our case, PCM, MFCM, and HPCM.
- Since the updating formulas for latent factors of users, queries and documents are dependent on one another, we could apply Stochastic Gradient Descent (SGD) algorithm to learn hidden factors during each EM iteration after  $\alpha_{uqd}$  is updated, instead of direct Least Squared estimation.

So far, we have illustrated our model as a general personalization framework for click models with probabilistic implications and explained its ability to connect the domains of users, queries and documents through latent factors.

## 4. EXPERIMENTS

To illustrate our experiments in this section, we begin with our data preparation and experiment set-up. We then discuss our evaluation metric. Finally, we compare our models

to a baseline model, where we study the impact of varying variables such as the number of factors and the number of iterations.

## 4.1 Experiment Setting

We sampled the search sessions used to train and evaluate click models from a commercial search engine in the U.S. market in English language in the first half of May 2011. A session consists of a input query, a list of returned documents on the search result page, a list of clicked positions, and a cookie ID to implicitly represent a user. We collected sessions subject to the following constraints: (1) the search session is on the first result page returned by the search engine; (2) all clicks in the session are on the search results, neither on sponsored ads nor on other web elements. We sort search sessions according to time stamps when the query is sent to the search engine and split them into training and the testing sets by a ratio of 1:1. In total, we collected approximately 66 million sessions over 2,264,889 distinct queries submitted by 2,975,306 different users. The total number of documents returned is 24,864,764. We filtered high-frequency queries and spam users that may dominate the training process. The average number of users per query is 4.253, and the average number of queries submitted by one user is 2.228. The detailed distribution of our data is presented in Table 1.

The column of “Percentage of New Query-User-Doc Tuple” denotes the percentage of the query-user-doc tuples that occur in testing dataset but are absent in training dataset. So does the last column for new query-doc pairs. As Table 1 shows, the percentage of these new pairs or tuples is rather high in real log data (28.90% for pairs and 67.75% for tuples). However, the baseline model can not predict well on these new pairs/tuples that it never meets in training dataset. We expect that our proposed model can boost up the performance on these pairs/tuples with help of collaborative filtering technology.

The baseline click model used in this paper for comparison is UBM (Section 2.3). We adopted EM inference process for parameters  $\alpha$  and  $\beta$  [8]. To avoid in-finite values in log-likelihood, a lower bound of 0.01 and an upper bound of 0.99 are applied to document relevance estimators for UBM. The number of iterations for EM algorithm is set as 100. To compare with UBM, we setup our factorization framework by using UBM to depict the part of position bias (Figure 1).

All experiments were carried out with HPC cluster of 144 machines, each with eight 2.4GHz cores and 16GB RAM.

## 4.2 Evaluation of Performance

To define and quantify the performance of a model, as well as to compare the soundness of different models, we introduce log-likelihood and perplexity as evaluation metrics in this section. These widely adopted metrics ([24, 3, 11, 12, 5, 23]) are used throughout our experiments. We then report the performance of our models, along with UBM baseline, under the according evaluation measurement.

### 4.2.1 Log Likelihood

Log Likelihood ( $LL$ ) is widely utilized in CTR prediction. One natural reason is that models often try to maximize the log likelihood of the joint distribution of click events in the course of inference. Therefore,  $LL$  becomes a direct measurement of the effectiveness of the estimators. For a

single document, the  $LL$  value is the log of the predicted clicking probability if there is a click on the document. The value is the log of the predicted non-clicking probability, otherwise. Take the Position Model as an example.  $LL_{\Theta}(\mathbf{S})$  is calculated as

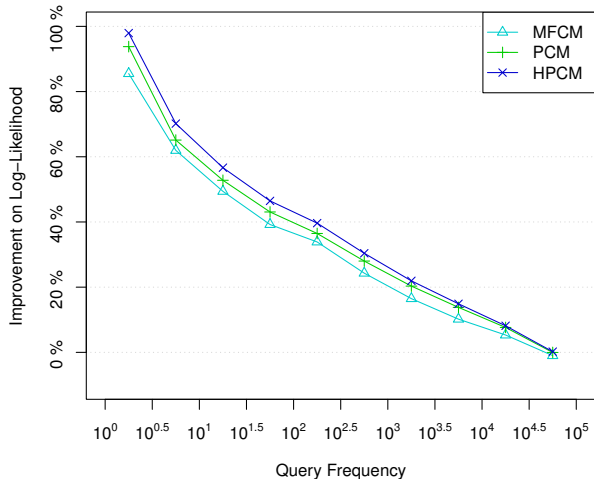
$$\frac{1}{S} \sum_{q,d,i} [S_{qd,i}^{\bullet} \log_2(\alpha_{qd}\beta_i) + S_{qd,i}^{\circ} \log_2(1 - \alpha_{qd}\beta_i)]$$

It is clear that a perfectly-fit set of estimators has a  $LL$  of value 0, and the smaller this value, the worse fit of the estimators. Also noteworthy is that when two models, or two sets of estimators, are to be compared, the improvement of  $LL_{\Theta}$  over  $LL_{\Theta'}$  is calculated up to an exponential transformation,  $(2^{LL_{\Theta} - LL_{\Theta'}} - 1) \times 100\%$ .

Model	Log-Likelihood	Improvement over UBM
UBM	-0.4236	-
MFCM	-0.3055	8.53%
PCM	-0.2577	12.18%
HPCM	-0.2448	13.20%

**Table 2: Performance of models measured by log-likelihood.**

We present our experiment results on all three proposed models (MFCM, PCM, and HPCM) and baseline UBM in Table 2. The average  $LL$  is  $-0.4236$  for UBM over all query sessions. Thanks to latent factor analysis, MFCM has a  $LL$  of  $-0.3055$  and scores a 8.53% improvement. With  $LL$  at  $-0.2577$  and  $-0.2448$ , our two personalized click model, PCM and HPCM, show improvements of 12.18% and 13.20% respectively. As expected, our HPCM leads to the most advance over UBM compared with other proposed models in our paper.



**Figure 3: Log-likelihood on test dataset with respect to different query frequencies. The factor models have more significant improvement ratios for tail queries.**

It has long been known that previous click models suffer from the so-called *tail problem*, whereby the click-through-rate prediction is far from satisfactory for infrequent queries.



Query Frequency	#Session	# Query	# User	# Document	Percentage of New Query-User-Doc Tuple	Percentage of New Query-Doc Pair
$10^0$ to $10^{0.5}$	14,935,135	1,077,691	2,211,723	11,747,545	83.47%	36.49%
$10^{0.5}$ to $10^1$	9,480,325	586,620	2,186,747	7,382,319	59.34%	22.12%
$10^1$ to $10^{1.5}$	9,030,565	283,449	290,581	4,171,647	50.79%	20.33%
$10^{1.5}$ to $10^2$	9,653,622	191,555	299,268	3,674,129	45.25%	23.21%
$10^2$ to $10^{2.5}$	7,573,039	64,186	142,140	1,708,899	46.00%	26.43%
$10^{2.5}$ to $10^3$	7,953,625	45,320	97,588	1,637,040	44.47%	21.31%
$10^3$ to $10^4$	4,958,208	13,644	86,969	637,606	40.36%	18.52%
$10^4$ to $10^{4.5}$	1,937,258	2,223	31,403	112,575	37.45%	11.20%
Total	66,547,897	2,264,889	2,975,306	24,864,764	67.75%	28.90%

Table 1: Summary of the dataset sampled from half month of click logs.

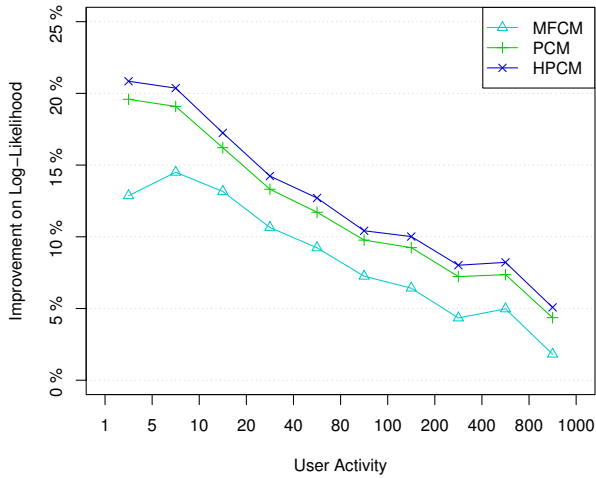


Figure 4: Log-likelihood on test dataset for various degree of user activeness. Our models have larger enhancement for infrequent users.

Our factor models solve the tail problem collaboratively. In Figure 3, we present the model improvement over baseline versus query frequency. The horizontal axis shows the log-scale of query frequency in training dataset. The improvement for rare queries boosts up to 80% to 100%. Similar phenomenon can be observed in Figure 4, where horizontal axis becomes the degree of user activeness characterized by user frequency. When both query frequency and user activeness decrease, the enhancement of our factor models shrinks, for UBM baseline can handle recurring training data well and leaves extremely limited space for other models to build up.

#### 4.2.2 Click Perplexity

Perplexity of click events is defined at each position independently rather than on the whole set. The metric is related to log-likelihood by putting negative of the latter to the exponent. In Position Model, for instance, click Perplexity ( $p$ ) at position  $i$  is defined as  $p_{\Theta}^i(\mathbf{S}) =$

$$2^{-\frac{1}{S_i} \sum_{q,d} [S_{qd,i}^* \log_2(\alpha_{qd}\beta_i) + S_{qd,i}^0 \log_2(1-\alpha_{qd}\beta_i)]}$$

Smaller perplexity indicates higher prediction quality, and the optimal value is 1. When comparing two models, the improvement of perplexity value  $p_{\Theta}^i$  over  $p_{\Theta'}^i$  is given by

$$\frac{p_{\Theta'}^i - p_{\Theta}^i}{p_{\Theta'}^i - 1} \times 100\%$$

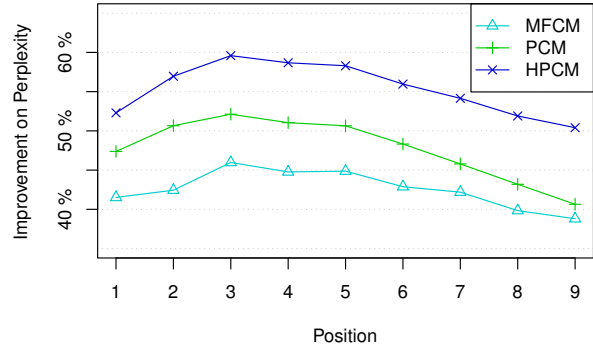
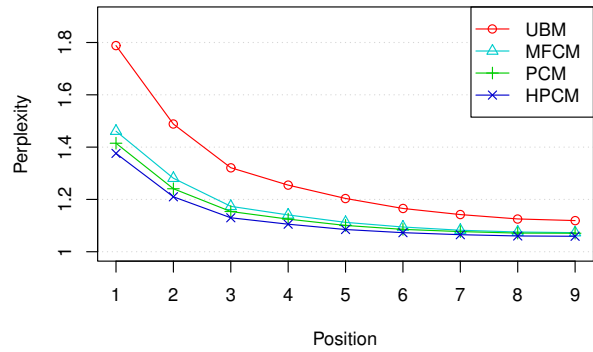


Figure 5: (a) upper. Model performance measured by perplexity at each position. (b) lower. Model improvement ratios over UBM at each position.

The experiment results of all three factor models and the baseline are summarized in Figure 5(a). The average click perplexity over all query sessions and positions is 1.1659 for MFCM and 1.1488 for PCM, 42.74% and 48.65% improvements per position over UBM (1.2898) respectively. Again, HPCM outperforms the other two factor models significantly, scoring an average click perplexity of 1.1293, which

enhances 55.39% of UBM. As shown in Figure 5(a), the improvement ratios of all proposed models in this paper does not suffer greatly when going down the SERP. The improvement of HPCM over UBM even hits to 50% for the lowest position. The figure also demonstrates that click prediction quality of HPCM is better than other models at every position. MFCM cannot compete with PCM or HPCM because of its lack of personalization, while PCM can be too over-fit and sacrifice performance at higher positions where user diversity is not that dominant. These results are consistent with our reasoning in Section 3.

### 4.3 Impact of Model Parameters

The three factor models we proposed in Section 3 have two model parameters, the number of factors and the number of iterations for SGD algorithm to well converge. We discuss the influence of these two parameters on model performance.

#### 4.3.1 Number of Factors

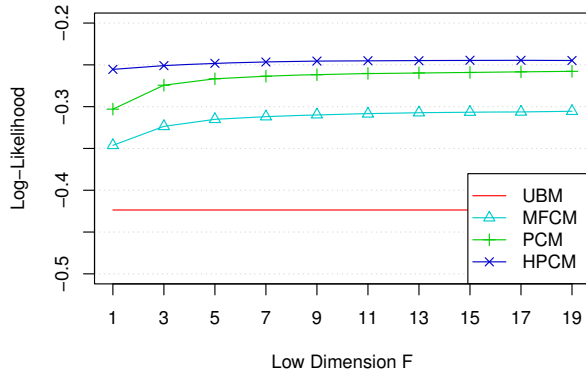


Figure 6: The influence of the number of factors on performance under log-likelihood. A few number of factors are adequate for stable performance.

We study the influence of the number of factors on all of our three proposed models, MFCM, PCM and HPCM, and present the experiment results in Figure 6. MFCM needs more than 10 factors to achieve a relatively stable performance, while PCM needs around 8, for PCM has one more domain of user than MFCM and the number of parameters to be estimated is greater. This also explains the reason why HPCM requires fewer number of factors to stabilize than PCM. Despite of the disparity among the models, the number of factors needed is rather low in general, implying that a few latent factors are adequate to catch most features of users, queries and documents.

#### 4.3.2 Number of Iterations

To infer the factor models, we need to run SGD algorithm iteratively in one EM iteration. We study whether our inference procedures converge and how fast they converge on our datasets. Figure 7 gives the curve showing the relationship between model performance and the number of iterations for factor models. All of our factor models converge. It is presented that more than 9 iterations of factorization are needed for MFCM to catch up with the performance of our baseline UBM. The tensor models, PCM and HPCM, become stable after around 20 iterations, while outperforming UBM from the beginning of iterations.

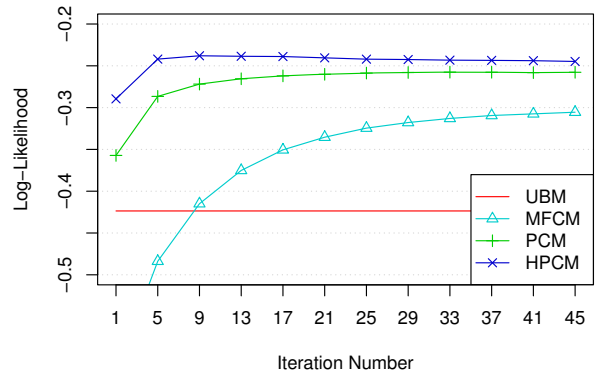


Figure 7: The influence of the number of iterations on performance under log-likelihood. All our models converge as the number of iterations goes up.

## 5. DISCUSSION

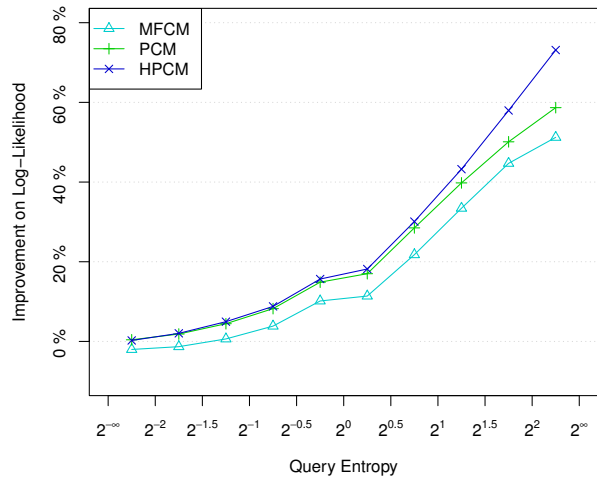


Figure 8: Perplexity on the test data for different query entropies. Factor models advance more when query entropy is higher, signaling supremacy on informational queries

We have demonstrated the sound performances of three factor models, MFCM, PCM and HPCM, in previous section, we then look into the enhancement and discuss our insights during the experiments.

Define query entropy of query  $q$  as

$$H(q) = - \sum_d CTR_{q,d} \log_2 CTR_{q,d}$$

where  $CTR_{q,d}$  is the predicted click-through-rate for document  $d$  under query  $q$ .

A high entropy signals a higher degree of complexity, or more precisely unpredictability. We present model improvement on Log-likelihood as query entropy varies in Figure 8. The horizontal axis is query entropy in log-scale. Although it is clear that the improvement is insignificant for low entropy queries, which mainly consist of navigational queries



like `amazon.com` and user diversity is negligible, the enhancement on performance impressively boosts up for more than 60% as query entropy rises. The reason is that high entropy queries possess complicated click logs that existing models cannot effectively deal with, whereas collaborative filtering can solve complex and diverse clicks efficiently. This matches our motivations and expectations of our factor models.

## 6. CONCLUSION

In this paper, we propose three factorization click models, which include: (1) a matrix factorization click model (MFCM) that relates queries and documents collaboratively; (2) a personalized click model (PCM) that uses indiscriminate tensor factorization to exploit the latent relationships of users, queries and documents; (3) a hybrid personalized click model (HPCM) with an emphasis on query-document interactions while characterizing user variations simultaneously.

We carry out a set of extensive experiments with two evaluation metrics on a real-world dataset. Our results clearly show the enhancement over state-of-the-art results. We see that MFCM performs well for capturing latent feature vectors of queries and documents, while PCM is better for the capability of personalization. Furthermore, HPCM achieves the greatest improvement by combining the strengths of previous two models. In addition, we show that our models are able to handle tail queries and informational queries, both of which are limitations in previously existing click models.

Through factor models, we address the personalization issues by characterizing user preferences toward documents. To simplify the model, however, we assume constant user preferences that are not varying with time. A future direction is to extend our models to a temporal click model to incorporate user preference dynamics to click models.

## 7. ACKNOWLEDGMENTS

The authors would like to thank Zhepeng Yan, Nan Liu, Bin Cao and Wenchen Zheng for their help and efforts.

This work was supported by Hong Kong RGC GRF project 621010, the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, and the National Natural Science Foundation of China Grant 61033001, 61061130540, 61073174.

## 8. REFERENCES

- [1] D. Agarwal, R. Agrawal, R. Khanna, and N. Kota. Estimating rates of rare events with multiple hierarchies through scalable log-linear models. In *KDD*, 2010.
- [2] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR*, 2006.
- [3] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *WWW*, 2009.
- [4] W. Chen, Z. Ji, S. Shen, and Q. Yang. A whole page click model to better interpret search engine click data. In *AAAI*, 2011.
- [5] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *WSDM*, 2008.
- [6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1), 1977.
- [7] G. Dupret and C. Liao. A model to estimate intrinsic document relevance from the clickthrough logs of a web search engine. In *WSDM*, 2010.
- [8] G. E. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. In *SIGIR*, 2008.
- [9] L. A. Granka, T. Joachims, and G. Gay. Eye-tracking analysis of user behavior in www search. In *SIGIR*, 2004.
- [10] F. Guo, C. Liu, A. Kannan, T. Minka, M. Taylor, Y.-M. Wang, and C. Faloutsos. Click chain model in web search. In *WWW*, 2009.
- [11] F. Guo, C. Liu, and Y. M. Wang. Efficient multiple-click models in web search. In *WSDM*, 2009.
- [12] B. Hu, Y. Zhang, W. Chen, G. Wang, and Q. Yang. Characterizing search intent diversity into click models. In *WWW*, 2011.
- [13] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, 2002.
- [14] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR*, 2005.
- [15] D. Kempe and M. Mahdian. A cascade model for externalities in sponsored search. Springer-Verlag.
- [16] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, August 2009.
- [17] C. Liu, F. Guo, and C. Faloutsos. Bbm: bayesian browsing model from petabyte-scale data. In *KDD*, 2009.
- [18] S. Rendle and S.-T. Lars. Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM*, 2010.
- [19] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *WWW*, 2007.
- [20] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, 2008.
- [21] W. Xu, E. Manavoglu, and E. Cantu-Paz. Temporal click model for sponsored search. In *SIGIR*, 2010.
- [22] Y. Zhang, W. Chen, D. Wang, and Q. Yang. User-click modeling for understanding and predicting search-behavior. In *KDD*, 2011.
- [23] Y. Zhang, D. Wang, G. Wang, W. Chen, Z. Zhang, B. Hu, and L. Zhang. Learning click models via probit bayesian inference. In *CIKM*, 2010.
- [24] Z. A. Zhu, W. Chen, T. Minka, C. Zhu, and Z. Chen. A novel click model and its applications to online advertising. In *WSDM*, 2010.

## APPENDIX

The resemblance of the log-likelihood of PCM and previous click models can be easily observed, so similar EM algorithm can be used to infer our new model. We explain the two steps separately and give the updating equations for a simpler case, where the number of latent vectors is one for all users, queries and documents.

## E Step

Let us define  $\mathbf{E}$  as the unobserved events of examination. EM algorithm first finds the expected value of the log likelihood  $\log P(\mathbf{S}, \mathbf{E}|\Theta)$  with respect to unobserved  $\mathbf{E}$  given observed  $\mathbf{S}$  and current parameters  $\Theta$ . That is,

$$\begin{aligned} Q(\Theta, \Theta^{t-1}) &= E[\log P(\mathbf{S}, \mathbf{E}|\Theta)|\mathbf{S}, \Theta^{t-1}] \\ &= \sum_E \log P(\mathbf{S}, E|\Theta) \times P(E|\mathbf{S}, \Theta^{t-1}) \\ &= \sum_{u,q,d,i} \sum_{C,E} [\log P(C, E|u, q, d, i, \Theta) \\ &\quad \times P(E|C, u, q, d, i, \Theta^{t-1})] \\ &= \sum_{u,q,d,i} \sum_{C,E} [\log P(C, E|\Theta) P(E|C, \Theta^{t-1})] \end{aligned}$$

where  $\Theta$  is the set of parameters following the tradition. We drop  $\{u, q, d, i\}$  temporarily for notation convenience.

Note that when the number of factors is one,  $U_u$ ,  $Q_q$  and  $D_d$  all become real numbers. By the assumption of the Position Model, the probability  $P(E|\mathbf{S}, \Theta^{t-1})$  mentioned above can be simplified to different cases.

$$\begin{aligned} P(E = 1|C = 1, \Theta^{t-1}) &= 1 \\ P(E = 0|C = 1, \Theta^{t-1}) &= 0 \\ P(E = 1|C = 0, \Theta^{t-1}) &= \frac{\beta_i^{t-1}(1 - U_u^{t-1}Q_q^{t-1}D_d^{t-1})}{1 - U_u^{t-1}Q_q^{t-1}D_d^{t-1}\beta_i^{t-1}} \\ &\doteq \tilde{C}_{uqd,i}^{t-1} \\ P(E = 0|C = 0, \Theta^{t-1}) &= 1 - \tilde{C}_{uqd,i}^{t-1} \end{aligned}$$

where  $\tilde{C}_{uqd,i}^{t-1}$  can be interpreted as the probability of document  $d$  being *irrelevant* of query  $q$  judged by certain user  $u$ .

$P(C, E|\Theta)$  is even more straight-forward, hence omitted. Then  $Q(\Theta, \Theta^{t-1})$  can be computed as

$$\begin{aligned} Q(\Theta, \Theta^{i-1}) &= \sum_{\mathbf{S}^\bullet} \{ \log(\alpha_{uqd}\beta_i) - \frac{1}{2\sigma^2}(\alpha - U_u Q_q D_{d_i})^2 \\ &\quad - \frac{1}{2\sigma_U^2}U_u^2 - \frac{1}{2\sigma_Q^2}Q_q^2 - \frac{1}{2\sigma_D^2}D_{d_i}^2 \} \\ &\quad + \sum_{\mathbf{S}^\circ} \{ \log((1 - \alpha_{uqd})\beta_i) \\ &\quad - \frac{1}{2\sigma^2}(\alpha - U_u Q_q D_{d_i})^2 - \frac{1}{2\sigma_U^2}U_u^2 \\ &\quad - \frac{1}{2\sigma_Q^2}Q_q^2 - \frac{1}{2\sigma_D^2}D_{d_i}^2 \} \tilde{C}_{uqd,i}^{t-1} \\ &\quad + \log(1 - \beta_i)(1 - \tilde{C}_{uqd,i}^{t-1}) \} \end{aligned}$$

This completes the Expectation step.

## M Step

The M step of EM iteration tries to maximize the expectation computed above, that is, to find

$$\Theta^t = \operatorname{argmax}_{\Theta} Q(\Theta, \Theta^{t-1})$$

Taking derivatives of  $Q(\Theta, \Theta^{t-1})$  in respect to  $\Theta$  produces

the updating formulas for parameters.

$$\begin{aligned} \beta_i^t &= \frac{S_i^\bullet + \sum_{uqd} (S_{uqd}^\circ \tilde{C}_{uqd,i}^{t-1})}{S_i} \\ U_u^t &= \frac{\frac{1}{\sigma^2} \sum_{q,d} (\alpha_{uqd} Q_q D_d W_{uqd}^{t-1})}{\frac{1}{\sigma^2} \sum_{q,d} Q_q^2 D_d^2 W_{uqd}^{t-1} + \frac{1}{\sigma_U^2} \sum_{q,d} W_{uqd}^{t-1}} \\ Q_q^t &= \frac{\frac{1}{\sigma^2} \sum_{u,d} (\alpha_{uqd} U_u D_d W_{uqd}^{t-1})}{\frac{1}{\sigma^2} \sum_{u,d} U_u^2 D_d^2 W_{uqd}^{t-1} + \frac{1}{\sigma_Q^2} \sum_{u,d} W_{uqd}^{t-1}} \\ D_d^t &= \frac{\frac{1}{\sigma^2} \sum_{u,q} (\alpha_{uqd} U_u Q_q W_{uqd}^{t-1})}{\frac{1}{\sigma^2} \sum_{u,q} U_u^2 Q_q^2 W_{uqd}^{t-1} + \frac{1}{\sigma_D^2} \sum_{u,q} W_{uqd}^{t-1}} \end{aligned}$$

where

$$W_{uqd}^{t-1} = \sum_{\mathbf{S}_{uqd}^\bullet} 1 + \sum_{\mathbf{S}_{uqd}^\circ} \tilde{C}_{uqd,i}^{t-1}$$

can be viewed as a fixed weighting term for each element in tensor.  $\alpha$ , however, does not have a close form calculation, but an equation to be solved numerically. The inference of EM algorithm completes. Each iteration is sure to increase log likelihood and the process is bound to converge to a local maximum of the joint likelihood function.