Simple Personalized Search Based on Long-Term Behavioral Signals

Anna Sepliarskaia[1], Filip Radlinski[2]*, and Maarten de Rijke[1]

[1] University of Amsterdam, Amsterdam, The Netherlands
{a.sepliarskaia,derijke}@uva.nl
[2] Google, London, United Kingdom
filiprad@google.com

Abstract. Extensive research has shown that content-based Web result ranking can be significantly improved by considering personal behavioral signals (such as past queries) and global behavioral signals (such as global click frequencies). In this work we present a new approach to incorporating click behavior into document ranking, using ideas of click models as well as learning to rank. We show that by training a click model with pairwise loss, as is done in ranking problems, our approach achieves personalized reranking performance comparable to the state-of-the-art while eliminating much of the complexity required by previous models. This contrasts with other approaches that rely on complex feature engineering.

1 Introduction

Search engines today combine numerous types of features when producing a ranking for a given query. They must provide ranked lists of results that are relevant (based on content), engaging (based on past user engagement), timely, and personally of interest to the user. These competing goals have led to a vast amount of work on each of them. Our focus is on personalization, which involves reranking documents on the search engine result page (SERP) so as to better satisfy a particular user's information need.

We present a novel approach to personalize search results with a model that is as effective as current state-of-the-art approaches, yet much simpler. By starting with a ranking produced by a commercial search engine, we know that the content of the top retrieved results is already likely to be of high relevance. However, we observe that usage still differentiates users and use this fact to rerank retrieval results based on implicitly collected usage. Consider, for instance, queries with only one intent but with a wide variety of relevant links such as "information retrieval conference." Links to SIGIR, ECIR, ICTIR, as well as links to general information on conferences are likely to be relevant. But each user has her own conference preference, which the system can infer from the user's past behavior—even if the user may be unable to formulate this preference directly in a query.

Previous research on personalizing search using behavioral data has found that to improve the ranking for a given user, information from the user's short-term and long-term behavior can be used [1, 4, 17]. Here, short term behavior is information from the

[*] Work done while this author was at Microsoft, UK.

session in which the user is currently engaged; long-term behavior concerns information from all of the user's search history. We focus on the use of long-term behavior for personalizing search as long-term behavioral signals have led to larger improvements than short-term behavioral signals [1, 18]. Also, short-term features cannot be used for the first query of a session, and over 40% of all sessions are of this sort [17].

At a high level, our approach calculates document scores given a query issued by a user, for each document d in the SERP. The score is a simple function combining three components: how well the document matches the query, how likely the user is to engage with documents at a given position, and how likely a user is to engage with a particular document. Perhaps surprisingly, despite not relying on handcrafted rules or sophisticated feature engineering, we show that performance is competitive with state-of-the-art models. Thus our key contribution is to show that formulating the optimization problem in this way removes the necessity for previously published complexity. We anticipate that by learning a simpler model, personalize reranking becomes more generally applicable, less complex computationally, and less error prone.

2 Related work

There are several approaches to addressing personalized search, each with its own benefits and drawbacks. First, one needs to understand when reranking is needed. The distinction of queries in three types—navigational, informational and transactional—is well-known [2]. Users submitting navigational and transactional queries use search engines to retrieve easily findable and recognizable target results; for most navigational and transactional queries reranking is well understood [14, 21]. Teevan et al. [21] show an easy and low-risk Web search personalization approach for navigational queries. Their approach achieves more than 90% accuracy. However, it works on the small segment of queries that the same user has issued at least three times. Query ambiguity is one of the indicators to inform us about changing the order of documents. Features and measures to predict it are proposed in [20]. If multiple documents have a high probability of being clicked following the query, then there is a great potential to improve the ranker.

The second type of related work concerns click models. Click models use implicit feedback to predict the probability of clicks [7]. Clicks can be a good indicator of failure or success. Features from click models are very useful for ranking documents [10, 11]. However, few click models are personalized [16]. As click models use implicit feedback, manual assessment is not required nor is feature engineering. These models work well for improving the click through rate (CTR). However, to re-rank URLs the relative order of predicted relevance is more important than absolute CTR value [6]. The click model that achieves the best performance for predicting probability of click is the User Browsing Model (UBM) [8]. The main difference between UBM and other models is that UBM takes into account the distance from the current document to the last clicked document for determining the probability that the user continues browsing.

The third type of approach to behavior-based personalized search uses feature engineering to create behavior features and then learn a ranking function [13, 18, 22]. Work that follows this approach differs in the choice of machine learning algorithms used. LambdaMart [3] is used in [13]. Several learning-to-rank algorithms as well as regression models are used in [22]. Logistic regression is used in [18]. Cai et al. [4, 5] use matrix factorization and restrict themselves to users with a sufficient volume of interactions. All of them devote significant attention to feature engineering. For example, Masurel et al. [13] use the probability that the user skips, clicks or misses the documents. The winners of the 2014 Kaggle competition on personalized search use over 100 features [13].

3 Method

We begin by providing a general description of our personalized search method and the intuitions behind it. At a high level, our goal is to obtain a simple yet effective model. The simplicity is achieved by an easily interpretable function that scores documents. The document score reflects the probability that the document is relevant, which depends on three random variables: attractiveness of the document to the user, attractiveness of the document to the query and examination of the rank of the document. The uniqueness of our approach is that, in contrast to previous models, we do not optimize the log likelihood of click probability but explicitly fit the probability that one document is more relevant than another in the SERP.

Our method shares traits of learning to rank methods and click models. Inspired by approaches in non-personalized pairwise learning to rank, we explicitly model the probability that one document is more relevant than another one. As in click models, personalized reranking involves modeling the relevance of documents using historical personal interactions with them. Further, we propose to train our model using long-term behavioral signals, which can be compared with classical click models [6, 8, 12] in its simplicity and approach, but it is as effective as recent complex models.

In our algorithm, position bias is taken into account. We follow the position model [7], in which it is assumed that examination of URLs on a SERP is a function of their rank and does not depend on examinations and URLs at higher positions. However, we assume that examination also depends on the query. Moreover, we have a factor that reflects attractiveness of a document to a given query. None of these parameters are personalized, therefore, we introduce new ones that are user specific. We introduce only one type of user specific parameters in this paper—attractiveness of a document to a specific user—but others could easily be integrated in a similar fashion.

We first introduce some notation: (a) q denotes a query, r a rank, d a document, u a user; (b) $e_{q,r}$ denotes the examination of a document at rank r in a SERP produced for q; (c) $a_{q,d}$ is the attractiveness of document d for query q; (d) $a_{u,d}$ is the attractiveness of d for user u. We will use the sigmoid function

$$\sigma(x) = 1/(1 + \exp(-x))$$

and the indicator function

$$I(x) = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{if } x \text{ is false.} \end{cases}$$

Given a query q submitted by user u, and a (non-personalized) SERP produced in response to q, our model re-ranks a document d that is originally placed at rank r in the SERP using the following scoring function:

$$\mathit{score}(q, d, u, r) = \sigma(a_{q,d}) \cdot \sigma(e_{q,r}) \cdot \sigma(a_{u,d}). \tag{1}$$

The learned parameters of the proposed model are $a_{q,d}, e_{q,r}, a_{u,d}$, which are single numbers. We use the sigmoid function to map these parameters to a probability.

We instantiate our model by training it based on implicit feedback from users. Given a query and user, we assume that the label of a given document is given by how the user interacts with it (click on it)—described specifically in Section 4. To achieve comparable results with the state-of-the-art model, we take inspiration from learning to rank methods and predict pairwise preferences of documents. More precisely, we map each document in the SERP to a number and the greater the difference between these numbers the higher probability that one document is more relevant than another. Specifically, for a given tuple (query q and user u) each pair of URLs d_i and d_j in a SERP with different labels is chosen. For each such pair we compute the scores $s_i = score(q, d_i, u, i)$ and $s_j = score(q, d_j, u, j)$, by using the parameters $a_{q,d_i}, e_{q,r_i}, a_{u,d_i}, a_{q,d_j}, e_{q,r_j}, a_{u,d_j}$, that were received up to that step. Let $d_i \prec d_j$ denote the event that d_i should be ranked higher than d_j. The scores are mapped to a learned probability that d_i should be ranked higher than d_j via a sigmoid function:

$$p(d_i \prec d_j) = \sigma(s_i - s_j). \tag{2}$$

We use a gradient descent formulation to minimize the cross-entropy function for each pair of documents in the SERP:

$$C(d_i, d_j) = -I(d_i \prec d_j) \cdot \log(p(d_i \prec d_j)) - (1 - I(d_i \prec d_j)) \cdot \log(p(d_j \prec d_i)). \tag{3}$$

Our method consists of three phases: first it tunes $e_{q,r}$, then $a_{q,d}$, and finally $a_{u,d}$. At each step the training procedure uses stochastic gradient descent (SGD), sequentially scanning the list of SERPs, calculating the gradient of the loss function for a SERP as

$$C_{serp} = \sum_{d_i, d_j} C(d_i, d_j), \tag{4}$$

and updating parameters $p_{uqd_1}, \ldots, p_{uqd_{10}}$ according to the following equation:

$$p_{uqd_i} \mathrel{+}= \eta \cdot \frac{\partial C_{serp}}{\partial s_i} \cdot \frac{\partial s_i}{\partial p_{uqd_i}}, \tag{5}$$

where η is a SGD-step, and p_{uqd_i} is one of $e_{q,r}$, $a_{q,d}$, $a_{u,d}$, depending on the phase.

We refer to our reranking model as specified in this section as *personalized ranked attractiveness* (PRA).

4 Experiments

In this section, we compare PRA with state-of-the-art models for personalized reranking. For this purpose we use data from the Yandex Personalized Web Search challenge [23]. We begin by noting that this dataset is the only publicly available dataset that satisfies our experimental needs. It contains information about SERPs and historical interaction with all documents shown to users: documents with their ranks and clicks on them. It also provides information on which user issued the query and interacted with the SERP.

The Yandex Personalized Web Search challenge dataset is fully anonymized. There are only numeric IDs of users, queries, query terms, sessions, URLs and their domains. The dataset comes with a full description of the SERPs contained in it: (a) the query for which the SERP was generated; (b) the ID of the user who issued the query; (c) URLs with their ranks and domains; and (d) the user's interaction with documents on the SERP, that is, indicators of clicks on documents. In case of a click, the dwell time in time units is also included. The organizers of the challenge suggest that documents with a click and dwell times not shorter than 400 time units are highly relevant to the query [23]. The following preprocessing was performed on the dataset before release: (a) queries and users are sampled from only one region (a large city); (b) sessions containing queries with a commercial intent as detected with a proprietary classifier are removed; (c) sessions with top-K most popular queries are removed; the number K is not disclosed. Some key statistics of the dataset are: (a) number of unique queries: 21,073,569; (b) number of unique urls: 703,484,26; (c) number of unique users: 5,736,333; (d) number of sessions: 34,573,630; and (e) number of clicks in the training data: 64,693,054.

Participants in the challenge are asked to rerank documents in SERPs according to the users' personal preferences.

We infer labels of URLs using a common approach [24]: (a) a 0 (irrelevant) grade corresponds to documents with no clicks or clicks whose dwell time is less than 400 time units; (b) a 1 (relevant) grade corresponds to documents that are clicked with a dwell time of more than 400 time units or clicked documents that have the lowest rank from all clicked documents in the SERP. A *satisfied* click is a click with a dwell time of at least 400 time units. We use two popular binary evaluation metrics: Precision@1 (P@1) and MAP@10.

To assess the consistency of our results, we measure the performance of our algorithms on several days. The dataset covers a period of 27 days; we use the first 20 days for training and the last 7 days (days 21–27) for testing. For each test day, we train algorithms on all days prior to the test day, and evaluate on the data collected for the test day. We do this over seven days to verify that the day of the week does not affect performance.

4.1 Training PRA

Each time the algorithm scans a SERP, we call this a "step." We use several hyper parameters to train PRA: (a) We make 5 steps for tuning each of parameters $a_{u,d}$, $a_{q,d}$, $e_{q,r}$: first, the algorithm makes 5 steps for tuning $a_{u,d}$, then 5 steps for tuning $a_{q,d}$, and finally 5 steps for tuning $e_{q,r}$. (b) We learn PRA by SGD with decreasing learning rate. In each step the learning rate is equal to the reverse square root of the number of steps $learning\ rate = 1/\sqrt{step\ number}$ (c) At the beginning we initialize all parameters $a_{u,d}$, $a_{q,d}$, $e_{q,r}$ to zero.

4.2 Baselines

We consider several experimental conditions (to be described below) and several baselines. Two baselines are considered for all experimental conditions: (a) *ranker* (ORIG) –

the default order that search results were retrieved by the Yandex search engine; (b) *pointwise feature engineering* (PFE)—the winner of the Yandex Personalized Web Search challenge. The core of PFE [18] is feature engineering; it uses three types of feature. Some of the features reflect the basic ranker that feeds into the reranking: document rank, document id, query id, and so on. Another group of features describes the users' interactions with URLs: whether the user clicked, skipped or missed a document in the current session or the whole history. The third set of features are pairwise: they describe, for each pair of URLs in the SERP, which document has a higher rank. To train the PFE approach, Song [18] considers all queries and logistic regression as a classifier of satisfied clicks. (c) *User Browsing Model* (UBM) [8]—a click model that performs the best for prediction probability of click [9].

For some of our experimental conditions we consider additional baselines: (d) *past click on document* (PCLICK [21])—if the SERP contains a document that received a satisfied click from the user, then it is placed on the first rank; (e) *document click through rate* (DCTR)—rerank documents according to CTR for document-query pair.

4.3 Experimental conditions

In the literature, multiple experimental conditions have been considered for comparing approaches to personalized reranking. We consider the following: (a) *all queries*; (b) *rerank examined documents only*, where we consider all queries but with a truncated list of documents: documents below the lowest click are removed before running the evaluation; (c) *repeated document subset*: SERPs with documents that a person clicked on in the past; (d) *poor SERPs*; and (e) *cold start*, where we group users depending on the richness of their histories. We now describe those conditions in more detail.

All queries. For comparability with PFE we report results on the full set of queries in the dataset and the exact same parameters as were mentioned by Song [18].

Rerank examined documents only. To avoid falsely penalizing algorithms if they promote documents that are relevant but were not clicked simply because the user did not observe them, we also perform our experiments using all queries but with a truncated list of documents. Specifically, all documents below the lowest click are removed before running the evaluation. It is clear that SERPs with only the first retrieved document being clicked cannot be reranked in this condition, as all other documents are excluded for this particular analysis. To understand how the potential of algorithms to change the order of documents affects relative performance, we list the ranks of the lowest click in SERPs in the dataset in Table 1. In particular, note that after truncation, more than a half of the SERPs cannot be changed by any reranking algorithms. At the other end of the spectrum, for 2.5% of the SERPs, reranking algorithms can yield any permutation of the URLs in the originally retrieved list of results.

Repeated document subset. From previous studies [17, 21], we know that users' behavior on repeated queries is particularly predictable. People often try to re-find documents, which they have read before [19]. Therefore, we consider a third experimental condition: the set of SERPs with documents that a person clicked on in the past. More precisely, in order for a SERP to be included in this set it should contain one and only one previously clicked document, where a past click on the document may have been for a different

Table 1: Distribution of SERPs depending on the rank of the lowest click.

Rank of the lowest click	1	2	3	4	5	6	7	8	9	10
Percentage of SERPs	54.5	13.8	8.4	5.7	4.3	3.3	2.6	2.3	2.2	2.5

Table 2: Description of groups in the cold start problem.

Group number	1	2	3	4	5	6	7	8	9
Number of queries issued by users in the group	0	1–2	3–5	6–8	9–11	12–15	16–21	21–32	>32

query. This subset of SERPs contains 13.8% of the total. For this condition, we use PCLICK [19] as an additional baseline.

Poor SERPs. From [20] we know that reranking is best applied selectively. Query ambiguity is one of the indicators to inform us about changing the order of documents and a good model should not rerank subsets of documents on which the ranker works well. For most queries, the top ranked document is clicked substantially more often than any of the other documents. However, for more ambiguous queries, or queries where the ranking is particularly poor, this is not the case. To evaluate such queries, in this subset we include queries for which the top ranked document is clicked less than twice as often as the second ranked document. A total of 48% of the SERPs in the dataset satisfy this condition. We also consider an additional baseline for this experimental condition: GCTR, the global clickthrough rate as defined in Section 4.1.

Cold start. Naturally, there is the cold start problem: if a user or a query are new to the system, then it becomes more difficult to produce a proper ranking. To better understand the effectiveness of PRA we also provide information on the changes of algorithms' performance depending on the richness of users' histories. We divided users into nine groups depending on the number of sessions in their history in such a way that each group has about the same number of people, i.e., each group has roughly 11% of the users; see Table 2. The first group are the people that are new; group 2 contains users who issued one or two queries, etc. Below, we report experimental results per group.

5 Results

In this section we present our experimental results. We learned all models regardless of the experimental conditions. For each of the five experimental conditions defined above (all queries, examined documents only, repeated documents, query ambiguity and cold start problem), we report on the performance of our proposed approach, PRA, and of the baselines listed in Section 4.1.

5.1 All queries

Table 3 lists the results for the "all queries" condition. We see that the performance of PRA and UBM is comparable to that of PFE, the state-of-the-art. In terms of Precision@1

Table 3: Results for the “all queries” condition, on each test day: days 21–27.

		21	22	23	24	25	26	27
P@1	ORIG	0.597	0.596	0.588	0.596	0.594	0.587	0.581
	PFE	0.607	0.603	0.602	0.603	0.604	0.595	0.594
	UBM	0.603	0.600	0.596	0.600	0.600	0.591	0.587
	PRA	**0.612**	**0.610**	**0.604**	**0.611**	**0.607**	**0.600**	**0.597**
MAP	ORIG	0.719	0.718	0.713	0.718	0.714	0.712	0.709
	PFE	**0.726**	0.723	**0.723**	0.723	**0.724**	**0.718**	**0.717**
	UBM	0.724	0.724	0.719	0.724	0.722	0.717	0.713
	PRA	**0.726**	**0.725**	0.720	**0.725**	0.723	**0.718**	0.716

PRA always outperforms PFE and PFE outperforms UBM although the difference is not significant (t-test, p-value > 0.1). In terms of MAP, the difference between PFE, UBM and PRA is at most 0.3%, in either direction. All of these three models, PFE, UBM and PRA, significantly outperform ORIG, the production ranker (t-test, p-value < 0.01). Also, surprisingly, UBM has comparable performance with PFE (t-test, p-value > 0.1).

5.2 Rerank examined documents only

We turn to the second experimental condition, where models rerank only examined documents. First, as this query set excludes documents below the lowest clicked position from reranking, all algorithms achieve higher scores, as we can see by contrasting the results in Table 4 with those in Table 3. The scores for PFE and PRA in this experimental condition are higher than in the “all queries” condition, both in terms of Precision@1 and MAP. Second, PRA outperforms PFE and UBM on both metrics. The difference in terms of Precision@1 exceeds 1.5% for each day, sometimes reaching 2.3%. Also, PRA performs significantly better than PFE, the state-of-the-art, in terms of MAP (t-test, p-value < 0.01). PFE and UBM have comparable performance.

Observing the performance differences between PRA, PFE and UBM relative to Table 3 more carefully, we note that the performance of PRA improved more due to the filtering of unobserved results. This tells us that on the complete dataset PRA promoted more documents that were not observed by the user than PFE or UBM. Thus, while the results in Table 3 are conservative (assuming all documents below the lowest actual click to be not relevant), the results in Table 4 are optimistic (restricted to documents for which we have more reliable evaluation labels). In both cases, we find that PRA outperforms PFE and UBM. We expect that results from an online evaluation would be somewhere between these two bounds.

5.3 Repeated document subset

In this experimental condition we only consider SERPs that contain exactly one previously clicked document. As this segment of queries was the specific target of the method proposed by Teevan et al. [19], we consider the additional baseline PCLICK. Table 5 lists

Table 4: Results for the "rerank examined documents only" condition, on each test day: days 21–27.

		21	22	23	24	25	26	27
P@1	ORIG	0.597	0.596	0.588	0.596	0.594	0.587	0.581
	PFE	0.610	0.606	0.608	0.606	0.608	0.598	0.599
	UBM	0.610	0.600	0.606	0.613	0.610	0.600	0.597
	PRA	**0.628**	**0.627**	**0.620**	**0.629**	**0.627**	**0.621**	**0.623**
MAP	ORIG	0.719	0.718	0.713	0.718	0.717	0.712	0.709
	PFE	0.734	0.731	0.733	0.731	0.733	0.726	0.726
	UBM	0.730	0.730	0.728	0.731	0.732	0.726	0.723
	PRA	**0.741**	**0.740**	**0.735**	**0.741**	**0.740**	**0.736**	**0.737**

the results for this condition. PRA achieves the best overall Precision@1 scores, followed by PCLICK, PFE, UBM and ORIG. Note that the difference in performance between PRA and the other approaches is more than 1% on every single test day. Surprisingly, PCLICK significantly outperforms PFE (t-test, p-value < 0.01), even though PFE is far more complicated and includes features that reflect user interactions with documents.

Although UBM and PFE achieve a similar performance in other experimental conditions, in this one PFE achieves better results than UBM. This is a consequence of the fact that PFE is personalized and uses the whole history of a user to predict clicks. As expected, all approaches achieve better Precision@1 scores than ORIG.

Table 5: Results for the "repeated document subset" condition on each test day: days 21–27.

		21	22	23	24	25	26	27
P@1	ORIG	0.776	0.776	0.776	0.773	0.772	0.755	0.754
	PFE	0.819	0.801	0.825	0.800	0.817	0.782	0.805
	UBM	0.798	0.800	0.797	0.796	0.794	0.778	0.777
	PCLICK	0.839	0.838	0.838	0.836	0.830	0.817	0.815
	PRA	**0.851**	**0.849**	**0.848**	**0.848**	**0.842**	**0.830**	**0.831**
MAP	ORIG	0.850	0.850	0.850	0.849	0.848	0.836	0.835
	PFE	0.880	0.868	0.883	0.866	0.878	0.855	0.870
	UBM	0.866	0.867	0.866	0.865	0.864	0.853	0.853
	PCLICK	**0.894**	**0.893**	**0.893**	**0.892**	**0.888**	**0.879**	**0.880**
	PRA	0.893	0.891	0.891	0.891	0.886	0.877	**0.880**

Interestingly, the results for MAP show a different pattern. PCLICK and PRA work almost equally well: the difference between them is less than 0.3% and not statistically significant (t-test, p-value > 0.01). Both PCLICK and PRA perform significantly better

Table 6: **Results for the "poor SERPs" condition on each test day: days 21–27.**

		21	22	23	24	25	26	27
P@1	ORIG	0.420	0.415	0.415	0.415	0.425	0.424	0.424
	PFE	0.440	0.434	0.444	0.433	0.440	0.442	0.443
	UBM	0.440	0.435	0.437	0.437	0.440	0.444	0.443
	DCTR	0.450	0.448	0.433	0.450	0.446	0.441	0.443
	PRA	**0.460**	**0.458**	**0.458**	**0.458**	**0.457**	**0.456**	**0.458**
MAP	ORIG	0.617	0.614	0.611	0.614	0.618	0.617	0.618
	PFE	0.628	0.625	**0.630**	0.624	0.623	0.628	0.630
	UBM	0.627	0.627	0.623	0.628	0.624	0.630	0.630
	DCTR	0.628	0.626	0.610	0.627	0.621	0.617	0.620
	PRA	**0.635**	**0.633**	**0.630**	**0.634**	**0.632**	**0.629**	**0.633**

than PFE (t-test, p-value < 0.01), which is better UBM, which, in turn, significantly outperforms *ORIG*.

5.4 Poor SERPs

Here we present results on ambiguous queries or queries where the ranking is particularly poor with the additional baseline DCTR; see Section 4.1 for a more precise definition. Table 6 shows the results on this subset for Precision@1 and MAP. For both metrics PRA outperforms other approaches, followed by PFE, UBM, DCTR, and then ORIG. The difference between PRA and the other approaches is significant (t-test, p-value < 0.01). ORIG performs significantly worse than the other approaches, while for most test days the differences between PFE, UBM and DCTR are not significant.

Also, all algorithms work much better on the subset where the condition of *Poor SERPs* is not satisfied. The performances of ORIG, PFE, UBM and PRA are similar and the precision@1 scores are over 78%. To conclude, the PFE, UBM and PRA methods improve ambiguous queries, but do not affect non-ambiguous ones.

5.5 Cold start problem

In the "cold start problem" condition we provide information on the algorithms' quality depending on the richness of users' history. This experiment has several results; see Table 7 for the results for both Precision@1 and MAP.

First, despite the fact that ORIG is not personalized it performs better for users with a long history. One of the explanations of this is that people who use the search engine a lot learn to submit high quality queries [15]. Second, the personalized models PFE and PRA benefit more from a user's history than ORIG and UBM. For users who issued more than 32 queries the difference between ORIG and these model is more than 2% for both metrics. Also, for users with a limited history, PFE and UBM benefits more than other algorithms. However, for users with a rich history UBM performs worse than PFE, which in turn performs worse than PRA, but still much better than ORIG.

Table 7: Performance of algorithms depending on the number of queries issued by user.

		0	1–2	3–5	6–8	9–11	12–15	16–21	21–32	>32
P@1	ORIG	0.584	0.570	0.572	0.581	0.585	0.595	0.605	0.613	0.652
	PFE	**0.594**	**0.579**	0.581	0.592	0.597	0.608	0.620	0.631	0.684
	UBM	0.593	0.578	0.581	0.590	0.595	0.605	0.617	0.627	0.673
	PRA	0.588	0.577	**0.582**	**0.594**	**0.600**	**0.613**	**0.626**	**0.640**	**0.694**
MAP	ORIG	0.710	0.700	0.700	0.707	0.710	0.718	0.726	0.733	0.762
	PFE	0.714	0.702	0.704	0.712	0.717	0.725	0.734	0.744	0.783
	UBM	**0.715**	**0.705**	**0.706**	**0.714**	0.717	0.725	0.734	0.743	0.777
	PRA	0.710	0.700	0.703	0.713	**0.718**	**0.727**	**0.737**	**0.749**	**0.788**

6 Conclusion

As search engines often show ten documents as a result page, most users can find a relevant item among them. However, different users have different interests. Thus for some users the first document may be relevant, but for others not. Thus we study reranking documents according to user interest. We have proposed a new simple method for personalized search based on long-term behavioral signals that matches or outperforms the state-of-the-art for this task.

We note that current state of the art solutions are effective, however they require extensive feature engineering. The most effective approaches have more than one hundred features. The second approach for this problem is manually creating rules, which is bound to work on a small segment of queries only. Another approach is click models. Click models are a very elegant solution for this problem, but in several experimental conditions work significantly worse than the state of the art. In contrast, our algorithm is applicable to all result sets, does not require feature engineering, but has comparable performance in all experimental conditions. We achieve this performance by incorporating click models with learning to rank algorithms.

We compared our proposed method with the state-of-the-art and with manually defined rules using a publicly available data set. We considered multiple experimental conditions. In all conditions we perform as least as well as the state-of-the-art and in several conditions we significantly outperform it according to both metrics used, despite the simplicity of our method.

Finally, we observe that our proposed approach only covers queries that have been seen previously, in the training data. In the future we plan to extend our approach to previously unseen queries by incorporating query similarity in our model. Also we plan to incorporate different relevance signals from query results and behavioral facets (visited pages, eye movement, etc.)

Acknowledgements. This research was supported by the Microsoft Research Ph.D. program. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

Bibliography

[1] P. N. Bennett, R. W. White, W. Chu, S. T. Dumais, P. Bailey, F. Borisyuk, and X. Cui. Modeling the impact of short- and long-term behavior on search personalization. In *SIGIR*, pages 185–194, 2012.

[2] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

[3] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML*, pages 89–96, 2005.

[4] F. Cai, S. Liang, and M. de Rijke. Personalized document re-ranking based on bayesian probabilistic matrix factorization. In *SIGIR*. ACM, July 2014.

[5] F. Cai, S. Wang, and M. de Rijke. Behavior-based personalization in web search. *J. Assoc. for Inform. Sci. and Techn.*, 2017. To appear.

[6] O. Chapelle and Y. Zhang. A dynamic Bayesian network click model for web search ranking. In *WWW*, pages 1–10, 2009.

[7] A. Chuklin, I. Markov, and M. de Rijke. *Click Models for Web Search*. Morgan & Claypool Publishers, July 2015.

[8] G. Dupret and B. Piwowarski. User browsing model to predict search engine click data from past observations. In *SIGIR '08*, pages 331–338, 2008.

[9] A. Grotov, A. Chuklin, M. Ilya, L. Stout, F. Xumara, and M. de Rijke. A comparative study of click models for web search. In *CLEF, 2015*. Springer International Publishing, 2015.

[10] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, pages 133–142, 2002.

[11] T. Joachims. Evaluating retrieval peformance using clickthrough data. In *Text Mining*, 2003.

[12] C. Liu, F. Guo, and C. Faloutsos. BBM: Bayesian browsing model from petabyte-scale data. In *KDD*, pages 537–546, 2009.

[13] P. Masurel, K. Lefévre-Hasegawa, C. Bourguignat, and M. Scordia. Dataiku's solution to Yandex's personalized web search challenge. In *WSDM '14 workshop on web search click data*, 2014.

[14] N. Matthijs and F. Radlinski. Personalizing web search using long term browsing history. In *WSDM*, pages 25–34, 2011.

[15] H. Morgan, H. Claudia, and E. David. Learning by example: Training users with high-quality query suggestions. In *SIGIR*, New York, NY, USA, 2006. ACM.

[16] S. Shen, B. Hu, W. Chen, and Q. Yang. Personalized click model through collaborative filtering. In *WSDM*, pages 323–332, 2012.

[17] M. Shokouhi, R. W. White, P. Bennett, and F. Radlinski. Fighting search engine amnesia: reranking repeated results. In *SIGIR*, pages 273–282, 2013.

[18] G. Song. Point-wise approach for Yandex personalized web search challenge. In *WSDM '14 workshop on web search click data*, 2014.

[19] J. Teevan, E. Adar, R. Jones, and M. Potts. Information re-retrieval: repeat queries in Yahoo's logs. In *SIGIR*, pages 151–158, 2007.

[20] J. Teevan, S. T. Dumais, and D. J. Liebling. To personalize or not to personalize: modeling queries with variation in user intent. In *SIGIR*, pages 163–170, 2008.

[21] J. Teevan, D. Liebling, and G. R. Geetha. Understanding and predicting personal navigation. In *WSDM*, pages 85–94, 2011.

[22] M. Volkovs. Context models for web search personalization. In *WSDM '14 workshop on web search click data*, 2014.

[23] Yandex. Personalized web search challenge. http://www.kaggle.com/c/yandex-personalized-web-search-challenge/details/prizes, 2013.

[24] E. Yilmaz, M. Verma, N. Craswell, F. Radlinski, and P. Bailey. Relevance and effort: An analysis of document utility. In *CIKM*, pages 91–100, 2014.