Using Historical Click Data to Increase Interleaving Sensitivity

Eugene Kharitonov[†‡], Craig Macdonald[‡], Pavel Serdyukov[†], Iadh Ounis[‡]
[†]Yandex, Moscow, Russia
[‡] School of Computing Science, University of Glasgow, UK
[†]{kharitonov, pavser}@yandex-team.ru
[‡]{craig.macdonald, iadh.ounis}@glasgow.ac.uk

ABSTRACT

Interleaving is an online evaluation method to compare two alternative ranking functions based on the users' implicit feedback. In an interleaving experiment, the results from two ranking functions are merged in a single result list and presented to the users. The users' click feedback on the merged result list is analysed to derive preferences over the ranking functions. An important property of interleaving methods is their sensitivity, i.e. their ability to reliably derive the comparison outcome with a relatively small amount of user behaviour data. This allows testing of changes in the search engine ranking functions frequently and, as a result, rapid iterations in developing search quality improvements can be achieved.

In this paper we propose a novel approach to further improve interleaving sensitivity by using pre-experimental user behaviour data. In particular, the click history is used to train a click model, which is then used to predict which interleaved result pages are likely to contribute to the experiment outcome. The probabilities of presenting these interleaved result pages to the users are then optimised, such that the sensitivity of interleaving is maximised. In order to evaluate the proposed approach, we re-use data from six actual interleaving experiments, previously performed by a commercial search engine. Our results demonstrate that the proposed approach outperforms a state-of-the-art baseline, achieving up to a median of 48% reduction in the number of impressions for the same level of confidence.

Categories and Subject Descriptors: H.3.3 [Information Storage & Retrieval]: Information Search & Retrieval

Keywords: interleaving, online evaluation

1. INTRODUCTION

The evaluation of retrieval systems is vital to ensure progress in information retrieval (IR). Historically, the system-based evaluation with manual judgements [18] motivated by Cranfield's experiments has proven popular. However, this approach has several limitations, as discussed by Voorhees [18]. Firstly, often the manual judgements are expensive to collect, since labelling requires labour from the trained professionals. Secondly, document relevance can be hard to define, especially when the user's intent can be ambiguous or the result set is personalised [1]. On the other hand, the online user-based experimental methods, such as A/B testing (e.g., [17]) and interleaving [9, 10, 11, 14], can overcome these limitations. As these methods leverage the live query stream to infer the users' preferences, they are believed to represent the actual preferences of the users.

Interleaving was first proposed by Joachims et al. [10, 11] as an online unbiased evaluation method. The idea behind this method is the following. Given two ranking algorithms we want to compare, A and B, we run an interleaving experiment on a portion of the query stream. In a user session affected by the experiment, the results of A and B are mixed into an *interleaved* result list that is shown to the user just as a regular search result page. Next, the user's click behaviour is interpreted in order to derive which of the tested algorithms provides the users with a better results ranking.

Following Radlinski and Craswell [12, 13], we define the *sensitivity* of an interleaving method as its ability to derive reliable experiment outcomes with little data. High levels of sensitivity are important in web search retrieval for several reasons. Firstly, sensitive methods allow search engines to evaluate improvements and iterate fast, i.e. to make more business decisions in a unit of time and progress quickly. Secondly, with a sensitive method, it is possible to evaluate even subtle improvements in a short time. Finally, since a considerable part of the evaluated changes may lead to a degradation of the user experience, sensitive methods can help to identify such changes early and reduce the user frustration.

In this work, we concentrate on improving the sensitivity of the interleaving methods. As we will discuss in the next section, this problem has received a considerable attention from the research community. However, we address this problem from a new perspective. Our hypothesis is that by using the user behaviour data prior to the start of the experiment, it is possible to adjust how often a particular interleaved result list is shown, to achieve higher levels of sensitivity.

In order to test this hypothesis, we propose and evaluate a method to leverage historical user behaviour data observed in the click log to optimise the parameters of the interleaving experiment.

The contributions of this paper are three-fold:

- We propose a theoretically-motivated approach to increase the sensitivity of the interleaving experiments

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. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.
Copyright 2013 ACM 978-1-4503-2263-8/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505515.2505687.

by adjusting how often a particular interleaved result page is shown;

- We propose a user click model-based approach to predict the parameters of this optimisation problem before the experiment is performed;
- We perform a thorough experimental study of the proposed algorithms.

The remainder of the paper is organised as follows. After discussing the related work in Section 2, we derive a problem of optimisation of the interleaving experiment parameters in Section 3. Next, in Section 4 we discuss how this optimisation problem can be formulated and solved before deploying the experiment online. The dataset and the evaluation methodology used in our study are discussed in Sections 5 and 6, respectively. Further, we report our results in Section 7 and close the paper with conclusions and a discussion of the possible future work in Section 8.

2. RELATED WORK

We consider our work to be related with two areas of research. Firstly, the research in the area of the interleaving algorithms and, in particular, a variety of methods to improve sensitivity of interleaving is relevant to this work and we discuss it in Section 2.1. Secondly, in this paper, we rely on the recent progress in the user click modelling methods, discussed in Section 2.2.

2.1 Interleaving methods

Three popular interleaving methods were proposed so far: Balanced Interleaving [10], Team Draft [14], and Probabilistic Interleaving [9]. In addition, several modifications of these methods aimed to improve their sensitivity were considered [3, 12, 19]. While Radlinski and Craswell [12] aimed to study the agreement between various metrics used in Cranfield paradigm evaluations with outcomes of the online interleaving experiments, they also paid considerable attention to the problem of weighting clicks in the Team Draft algorithm to ensure the sensitivity of the experiments. They demonstrated that weighting the user clicks by the logarithm of the clicked document rank can lead to higher sensitivity of the interleaving experiments.

A similar idea was studied by Yue et al. [19], who proposed a method to train a credit assignment scheme so that the overall sensitivity of the interleaving is increased. More precisely, Yue at al. considered a supervised machine-learning problem where the outcome of the interleaving experiment known. Given the user behaviour data for these experiments, they learned a click weighting scheme so that the power of a statistical test on new experiments was maximised. Chapelle et al. [3] performed a thorough experimental comparative study of interleaving algorithms, concentrating on the Balanced Interleaving and Team Draft algorithms. In particular, they investigated relative sensitivity of these algorithms and several of their modifications with non-uniform click weighting schemes. For instance, their results suggested that ignoring clicks on the top results shared by both ranked lists considerably improved the sensitivity of the Team Draft algorithm.

As we can see from the above discussed work, a considerable body of previous research studied how to improve the sensitivity of the online experiments by means of adjusting the credit assignment (click weighting) scheme used in the corresponding interleaving algorithm. In our work, we consider a somewhat complimentary approach: given a particular credit assignment scheme, we investigate how one can control the probabilities of showing the interleaved result pages to the users to achieve maximum sensitivity. We expect that combining the previously proposed click weighting schemes with the approach considered in this work can lead to even higher levels of sensitivity. However we leave a thorough study of the possible combinations as a promising direction of future research.

Our proposed approach is based on the study of Radlinski and Craswell [13], who proposed a formal Optimised Interleaving framework describing how three components of an interleaving method (a set of interleaved result lists, a credit assignment scheme, and a distribution over interleaved result lists) can be combined so that the resulting interleaving algorithm is unbiased. Their proposed framework was used to build a family of interleaving algorithms with different credit assignment schemes. They also optimised the distribution over the interleaved result sets so that the uncertainty in a winner of a particular impression is maximised. This uncertainty is calculated with respect to a randomly clicking user. Despite relying on the framework proposed by Radlinski and Craswell [13], our approach has the following differences. Firstly, we study a different perspective of the interleaving sensitivity. Instead of considering the winner uncertainty within a single interleaved result list, we are aiming to reduce the number of user impressions that contribute little to the experiment outcome. Secondly, instead of considering a randomly clicking user, we leverage massive click log data containing the real user feedback and use it to train a click model that is further used to predict the actual behaviour of a real user.

Another closely related work is by Hofmann et al. [8]. In this work, the authors study the possibility to leverage the query log data to predict an interleaving experiment outcome without actually running the experiment. Their results suggests that Probabilistic Interleaving [9] can effectively re-use such historical data. In contrast, in our work we aim to leverage historical data to improve the sensitivity of the future interleaving experiments.

2.2 Click models

Apart from the research in the area of the interleaving methods, this paper is based on the recent progress in research on user behaviour modelling. The problem of interpreting and modelling the user's clicking behaviour is nontrivial, since the user's actions are prone to several biases. Arguably, *the position bias* is the most studied one [6]. It affects the way users click on results: results that are ranked higher collect more clicks from users even if they are not as relevant as the results ranked lower. One of the approaches to model the position bias is considered in the position click models [5, 15]. The underlying idea was formalised by Craswell et al. [5] in the *examination hypothesis*: a search result is clicked only if it is examined, and the user considers the result to be relevant. The position-based models assume that the examination probability depends only on the rank the result is presented at. A more sophisticated approach is considered by the cascade click model [5]. This model is based on the following assumption: a result can be examined only if all of the results ranked higher were examined. An important extension of the cascade model is the Dynamic Bayesian Network (DBN) click model later proposed by Chapelle and Zhang [4]: this model additionally accounts for the effect of the users abandoning their search, and is capable of modelling sessions with several clicks, by separating the perceived and the actual relevance of the results.

In most cases, the user click modelling has been used to extract new ranking features, or to substitute expensive manual assessment procedures [4]. In our work, the click modelling is used to predict the user behaviour once a new result page is shown to the user. A similar scenario was considered by Guo et al. [7] to evaluate click models by their ability to predict the position of the first and the last clicks.

A somehow related approach was used by Hofmann et al. [9] to evaluate interleaving algorithms: the parameters of the Dependent Click Model [7] were estimated from the available document relevance judgements, and the resulting click model was used to evaluate interleaving algorithms in the absence of an actual query log by generating a synthetic one. In contrast, in our work, we learn a user click behaviour model from the actual, not synthetic, query log data and hence leverage it for a completely different purpose: optimising an interleaving algorithm.

As can be seen from the above discussed work, the sensitivity of the interleaving methods attracted a considerable attention from the IR community. However, the possibility to use the historical user behaviour data to increase the interleaving sensitivity has not been studied before. After reviewing how interleaving experiments are performed, we describe the proposed interleaving approach in Section 3.

3. OPTIMISING INTERLEAVING SENSITIVITY

Before discussing the sensitivity of the interleaving experiments, we briefly review how the interleaving experiments are performed and introduce the required notation. Each interleaving algorithm essentially consists of three parts: a rule to build a set of the interleaved result lists for a query; a credit assignment scheme used to interpret the user clicks; and a distribution determining how often a particular interleaved result list is shown to the users. The latter is further referred to as the experiment *policy*. Further, assuming that a query q is fixed, let us introduce the notation used in this paper. L is a set of considered interleaved result lists, L_i stands for the ith interleaved result list, and $L_i(r)$ denotes a result ranked on the rth rank of L_i. Similarly to [3, 13], we define a credit function $\delta_i(r)$ that describes a score assigned to A ($\delta_i(r) > 0$) or B ($\delta_i(r) < 0$) after a user clicked on the result ranked on position r in the result list L_i.

A part of the user's interaction with a search engine result page, which starts with submitting a query and ends when the user submits a new query or leaves the search engine, is further referred to as *impression*. Each impression v is associated with a shown result page L_i and with the user's clicks. The interleaving credit scheme can be used to derive which of the compared systems (A or B) wins in a particular impression. More formally, the user's clicks observed in a particular impression v with a result list L_i shown are transformed into scores $h(v) = \sum_r \delta_i(r)\mathbb{1}\{r\ clicked\ in\ v\}$, where $\mathbb{1}\{\cdot\}$ is an indicator function. After that, an impression-level aggregation $c(\cdot)$ can be applied. For instance, this aggregation can represent which alternative (A or B) wins a particular impression (i.e. $c(v) = sign(h(v))$) or normalise the score by the number of clicks [3].

After running the experiment, a single statistic Δ is used to describe the experiment outcome [3]:

$$\Delta = \frac{1}{N}\sum_{j=1}^{N} c(v_j) \tag{1}$$

where N denotes the total number of impressions.

To simplify the analysis, we follow Yue et al. [19] and consider that A (B) wins in the interleaving experiment if it receives more credit than B (A). This approach is equivalent to considering an identity credit aggregation function $c(v)$. In order to represent the experiment outcome in this case, we define w^*_A to be the total credit assigned to A during the experiment. Similarly, w^*_B denotes the credit assigned to B. As a result, the experiment outcome can be defined as follows:

$$\Delta^* = \frac{w^*_A - w^*_B}{N} \tag{2}$$

Informally, Δ^* equates to the difference in the credits assigned to A and the credits assigned to B, divided by the total number of impressions the users observed in the experiment. Again, $\Delta^* > 0$ means that A outperforms B.

We hypothesise that organising the interleaving experiment policy in such a way that the result pages L_i that are unlikely to contribute to the difference in Equation (2) are shown as rarely as possible should improve the ability to derive reliable conclusions with less impressions. In other words, with N being fixed, the interleaved result pages that often lead to ties[1] should be shown less frequently in comparison with those that witness a contrast between A and B.

In the remainder of this section, we study how this goal can be achieved based on the Optimised Interleaving framework [13], initially assuming that information about the future user behaviour is available at the start of the experiment. We will relax this assumption in Section 4.

At first, let us denote the credit assigned to A after showing the result list L_i as w^{*i}_A. Similarly, we define w^{*i}_B. Given this notation, it is possible to expand w^*_A and w^*_B in the following way:

$$w^*_A = \sum_i w^{*i}_A;\ w^*_B = \sum_i w^{*i}_B \tag{3}$$

Putting (3) into (2) and further grouping the terms in the numerator, we re-write Δ^* as follows:

$$\Delta^* = \frac{\sum_i w^{*i}_A - \sum_i w^{*i}_B}{N} = \frac{\sum_i w^{*i}_A - w^{*i}_B}{N} \tag{4}$$

Next, we denote as N_i the number of impressions where L_i was shown and re-write (4) in the following way:

$$\Delta^* = \sum_i \frac{N_i}{N}\frac{w^{*i}_A - w^{*i}_B}{N_i} \tag{5}$$

Further, we refer to π_i as the probability of showing L_i in the interleaving experiment, i.e. the vector $\boldsymbol{\pi}$ represents the experiment policy. Substituting π_i, we obtain:

$$\Delta^* = \sum_i \pi_i \frac{w^{*i}_A - w^{*i}_B}{N_i} \tag{6}$$

Intuitively, with the total number of impressions N fixed, higher absolute values of Δ^* correspond to higher contrast between A and B, and lead to the ability to determine the experiment outcome with higher reliability. Thus, let us consider a simple upper bound on the absolute value of Δ^*:

[1] For instance, the result pages that do not attract clicks or often have similar credits assigned to both alternatives.

$$|\Delta^*| = \left|\sum_i \pi_i \frac{w_A^{*i} - w_B^{*i}}{N_i}\right| \le \sum_i \pi_i \frac{\left|w_A^{*i} - w_B^{*i}\right|}{N_i} \qquad (7)$$

Indeed, the absolute value of Δ^* is bounded by the product of the experiment policy and the statistics of the interleaved results lists $\frac{\left|w_A^{*i} - w_B^{*i}\right|}{N_i}$. This quantity is related to the contribution that the impressions with L_i make to the difference (2), as we discussed earlier.

So far, we have discussed the experiment outcome Δ^* and obtained the upper bound on its absolute value. However, this upper bound also provides us with an idea how the experiment policy can be adjusted before starting the experiment. Despite the fact that a higher upper bound does not necessary imply a higher value of Δ^*, in this work we argue that controlling π_i, so that the upper bound increases, leads to higher sensitivity of the interleaving. Intuitively, this idea can be expressed as follows: in a real world scenario, with everything else being equal, it is generally better not to show a result list L_i with a low value of $\frac{|w_A^{*i} - w_B^{*i}|}{N_i}$.

Having discussed the motivation behind our approach, let us consider the problem of selecting the optimal experiment policy before starting the experiment. Due to the random nature of user behaviour, it is reasonable to consider the *expected* difference in the credits assigned to A and B for a particular interleaved result list L_i, instead of $\frac{|w_A^{*i} - w_B^{*i}|}{N_i}$, which approaches this expectation as the number of impressions N_i grows. Further, we denote this quantity as μ_i:

$$\mu_i = \left|\lim_{N_i \to \infty}\left[\frac{w_A^{*i} - w_B^{*i}}{N_i}\right]\right| = \left|\mathbb{E}\left[C_A^i - C_B^i\right]\right| \qquad (8)$$

where C_A^i and C_B^i stand for the credits assigned to A and B after demonstrating the result list L_i to a user, respectively. The values of μ_i form the vector $\boldsymbol{\mu}$.

As discussed above, our goal is to adjust policy $\boldsymbol{\pi}$ so that the upper bound (7) increases. In order to achieve that, we leverage the Optimised Interleaving framework [13], which can be used to optimise the interleaving experiment properties without introducing biases. In particular, Radlinski and Craswell [13] define a criterion of the unbiased interleaving policy: the expected credit from a randomly clicking user should be zero. Formally, for an interleaving method to be unbiased, the following constraint on its set of interleaved result lists L, its credit function δ, and its policy $\boldsymbol{\pi}$ has to be met:

$$\forall k \; \sum_{i=1}^{|L|} \pi_i \sum_{r=1}^{k} \delta_i(r) = 0 \qquad (9)$$

Temporarily assuming that values of $\boldsymbol{\mu}$ are known and combining (7), (8), and (9) we formulate our optimisation problem as follows:

$$\boldsymbol{\mu}^T \boldsymbol{\pi} \to \max \qquad (10a)$$

$$\forall k \; \sum_{i=1}^{|L|} \pi_i \sum_{r=1}^{k} \delta_i(r) = 0 \qquad (10b)$$

$$\sum_i \pi_i = 1 \qquad (10c)$$

$$\forall i \; \pi_i \ge 0 \qquad (10d)$$

Indeed, the solution of the optimisation problem stated by the set of Equations (10) maximises the upper bound of the experiment's outcome $|\Delta^*|$ (10a), meets the fairness condition (10b) proposed by Radlinski and Craswell [13], and represents a valid distribution (10c & 10d). In this work, we argue that setting the interleaving experiment policy to the solution of the linear optimisation problem (10) leads to a higher sensitivity of the experiment.

Since the values of $\boldsymbol{\mu}$ are predicted from the noisy user feedback, this can cause undesired noise to the solution. Indeed, a small variation in values of $\boldsymbol{\mu}$ might result in the linear programming problem (10) having a completely different solution. In order to reduce this noise, we introduce a regularisation term to the optimisation objective (10a) that adds a penalty to solutions that diverge too far from the uniform policy $\boldsymbol{\pi}_U$[2]. Thus, we replace objective (10a) with the following expression:

$$\boldsymbol{\mu}^T \boldsymbol{\pi} - \alpha(\boldsymbol{\pi} - \boldsymbol{\pi}_U)^T(\boldsymbol{\pi} - \boldsymbol{\pi}_U) \to \max \qquad (10a^*)$$

where α is a non-negative scalar parameter. With α being zero (10a*) reduces to (10a), while large values of α force the solution to be the uniform vector. In the latter case, the pre-experimental user behaviour is ignored and all interleaved result pages are demonstrated to the users with equal probabilities. If L and δ coincide with that of the Team Draft algorithm, the uniform policy $\boldsymbol{\pi}_U$ corresponds to the policy of Team Draft, hence the solution of the optimisation problem (10) with large α coincides with the Team Draft algorithm. Therefore, Team Draft can be seen as a feasible solution of the optimisation problem (10), which does not rely on the user behaviour information and weights all possible interleaved result sets equally. On the other hand, with $\alpha = 0$ the optimal solution of (10) is completely defined by the noisy estimates of $\boldsymbol{\mu}$ from the previous user click behaviour and becomes "risky". A higher level of risk may result in higher improvements in the interleaving sensitivity, but it also may lead to a decrease in sensitivity when the estimates of $\boldsymbol{\mu}$ are incorrect due to prevalent noise in the user feedback.

We argue that the parameter α provides a convenient mechanism to control the amount of prior user feedback introduced into the resulting interleaving algorithm. For instance, α can be set to zero for queries that have sufficiently large pre-experimental user click data and set to infinity for queries with little user information available. However, in this work, we restrict α to be uniform for all queries, and leave the study of the per-query adaptation of α as a direction for future work.

Since the values of vector $\boldsymbol{\mu}$ are not known before starting the experiment, a question arises how $\boldsymbol{\mu}$ can be estimated. In this paper, we propose to use pre-experimental click log data to train a model of the user click behaviour and use it to *predict* $\boldsymbol{\mu}$. We discuss this approach in the next section.

4. USING THE PRE-EXPERIMENTAL DATA

As discussed in the previous section, once the values of $\boldsymbol{\mu}$ for the interleaved result pages L are available, it is possible to optimise the interleaving to achieve a higher sensitivity by selecting the solution of (10) as the experiment policy. In this section, we study how $\boldsymbol{\mu}$ can be estimated by using the click log data.

[2] In the experimental part of this paper, we work with the Team Draft-based set of interleaved result lists L and associated credit functions, so $\boldsymbol{\pi}_U$ is a feasible solution of (10). For other combinations of L and δ, other policies might be more suitable.

```
Input: Set of sessions $Q$; Beta prior parameters:
       $\alpha_a, \alpha_s, \beta_a, \beta_s$
Output: The click model parameters for each
        document $u$: $a_u$, $s_u$
$a_u^N \leftarrow 0$; $a_u^D \leftarrow 0$
$s_u^N \leftarrow 0$; $s_u^D \leftarrow 0$
foreach session $s \in Q$ do
    foreach result $u$ above or on the last clicked
    position do
        $a_u^D \leftarrow a_u^D + 1$
    end
    foreach clicked result $u$ do
        $a_u^N \leftarrow a_u^N + 1$
        $s_u^D \leftarrow s_u^D + 1$
    end
    $u \leftarrow$ last clicked document in $s$
    $s_u^N \leftarrow s_u^N + 1$
end
foreach $u$ do
    $a_u \leftarrow \frac{a_u^N + \alpha_a}{a_u^D + \alpha_a + \beta_a}$
    $s_u \leftarrow \frac{s_u^N + \alpha_s}{s_u^D + \alpha_s + \beta_s}$
end
```

Algorithm 1: Training the sDBN model, as described by Chapelle et al. [4].

We argue that leveraging the click log data is a very promising approach due to the following reason. Usually, only a tiny part of the query stream is used to perform an interleaving experiment, thus the whole query stream provides a considerably larger amount of data reflecting user preferences. Moreover, interleaving experiments are performed on a limited timescale, while commercial search engines can store query logs spanning several years of operation. As a result, one can easily obtain two orders of magnitude more impressions from the non-experimental log than what can be obtained from a whole experiment run for several weeks. We consider our goal to leverage this massive evidence to make the interleaving experiments more sensitive. The approach we propose achieves this goal in three steps. In the first step, a click model is trained on the user click behaviour history. In the next step, this pre-trained model is used to *predict* the future user behaviour and to estimate the parameters of the optimisation problem (10). Finally, we find the optimal policy and run the interleaving experiment on a part of the query stream.

In the remainder of this section, we discuss the click model (Section 4.1), the estimation of the optimisation objective parameters (Section 4.2), and a possible inter-query bias introduced (Section 4.3).

4.1 Training the click model

The general idea behind predicting the parameters $\boldsymbol{\mu}$ of the optimisation problem (10) is the following. Having observed a massive click log representing the users' behaviour, we can train a generative model of the user click patterns. Once the click model with the pre-trained parameters is available, it can be used to "explain" the behaviour observed in the click log and, more importantly, to model how users will behave once the result page is modified. The latter case is the most important in our task, since the interleaved result lists will often contain the same documents, but arranged differently. Also, these documents are likely to be presented in the deployed retrieval system before, but at different positions and surrounded by different documents. For this reason, we rely on the ability of the underlying click model to generalise while predicting the user behaviour.

As discussed in Section 2, a variety of generative click models have been proposed. In this work, we use a simple yet effective modification of the Dynamic Bayesian Network, simplified DBN (sDBN)[3], proposed by Chapelle and Zhang [4]. Informally, the sDBN model assumes that a user examines the result list from top to bottom. After examining a document u, the user either finds it attractive with probability a_u and clicks on it, or continues to the next document. After clicking on a document, the user is satisfied with probability s_u and stops the examination process. Thus, for a fixed query, the model has two parameters per document u: attractiveness a_u and the probability of satisfying the user s_u. These parameters are learned from a click log by means of Algorithm 1, described by Chapelle et al. [4]. This learning procedure imposes Beta priors on the model parameters and following [4] we set them to 1, i.e. $\alpha_a = \alpha_s = \beta_a = \beta_s = 1$.

After training the model parameters it can be used to predict $\boldsymbol{\mu}$ as we discuss in the next section.

4.2 Estimating the parameters

According to our definition (8), μ_i equates to the absolute value of the relative difference between credit assigned to alternatives A and B after observing infinitely many user interactions, or, alternatively to the absolute value of the expected difference in the credits assigned to A and B. Under the sDBN model, this quantity can be calculated analytically by means of Algorithm 2, as we prove in the following Lemma 1:

LEMMA 1. *Algorithm 2 calculates $\boldsymbol{\mu}$, defined in* (8).

PROOF. Consider an interleaved result list L_i and let $P_e(r)$ denote the probability of the user examining the position r, $P_c(r)$ denote the probability of clicking on position r by $P_c(r)$, and $P_s(r)$ denote the probability of the user being satisfied with the document in the rth position after clicking on it. Under this notation, we can re-write the expectation in (8) as follows:

$$\mathbb{E}\left[C_A^i - C_B^i\right] = \sum_r \delta_i(r) P_c(r) \tag{11}$$

Since the sDBN model assumes that the user clicks on a result only after examining it, we can express $P_c(r)$ in the following form:

$$P_c(r) = a_{u(r)} P_e(r) \tag{12}$$

In turn, under the cascade hypothesis the document in the rth position is examined only if the user is not satisfied with all the documents ranked above:

$$P_e(r) = \prod_{j<r} 1 - P_s(j) = \prod_{j<r} 1 - a_{u(r)} s_{u(r)} \tag{13}$$

Putting (13) and (12) in (11) we note that Algorithm 2 indeed calculates $\boldsymbol{\mu}$ as defined by Equations (8) & (11). □

4.3 Discussion

For queries for which there is no user click history available, the optimal policy can be reduced to the uniform policy

[3] We have also tried the Dependent Click Model [7] and found it to perform worse.

```
Input: Parameters of the click model, $a_u$, $s_u$; set of
       interleaved result lists, $L$
Output: Vector of the optimisation objective (10a*)
        parameters $\boldsymbol{\mu}$
//$P_e(r)$ denotes the probability of examining the $r$th
position
$P_e(1) \leftarrow 1$
foreach $L_i \in L$ do
    for $r \leftarrow 1$ to $|L_i|$ do
        $u \leftarrow L_i(r)$ // $u$ is the document on the $r$th
        position
        //expected credit from the $r$th position
        $\mu_i \leftarrow \mu_i + P_e(r)\delta_i(r)a_u$
        //probability of examining the next document
        $P_e(r+1) \leftarrow P_e(r)\,(1 - a_u s_u)$
    end
    $\mu_i \leftarrow |\mu_i|$
end
```

Algorithm 2: Estimating $\boldsymbol{\mu}$ with the pre-trained sDBN click model.

$\boldsymbol{\pi}_U$, which is equal to Team Draft's policy if the interleaved result lists and the credit function of Team Draft [14] are used. The question arises if improving sensitivity only for a part of queries (in our case, queries with the click data available) can introduce a bias into the interleaving experiment. The fairness criterion we use in the optimisation problem (10b) eliminates bias only within a single query and, to the best of our knowledge, there is no formal criteria that can define if an inter-query bias is introduced to an interleaving experiment. For instance, the experiment policy is optimised in [13], but the inter-query bias is not discussed. However, once formal criteria of the absence of the inter-query bias are proposed, it might be possible to add them to the optimisation problem constraints. Another possibility is to perform a long real-life study of the proposed algorithm's agreement with other means of the retrieval evaluation: system-based evaluations with manual judgements, A/B testing, and other interleaving methods.

However, in some cases the absence of the introduced bias can be guaranteed. For instance, the inter-query bias is not introduced if the tested change in the search engine ranker has equal chances to improve (degrade) the ranking quality for the frequent and long-tail queries.

5. DATASET

Interleaving is an online evaluation method. However, it is more convenient to evaluate the sensitivity of an interleaving method itself in an offline experimental setting, since this allows us to compare several methods on the same dataset. In this paper, we use an offline evaluation approach similar to the approach used by Radlinski and Craswell [13]. However, in our case two datasets are required. The first datasets represents the pre-experimental user behaviour and is used to train the click model parameters. It is further referred to as the *user modelling* dataset. The second dataset represents the interleaving experiments that were conducted as a part of the every-day experimentation practice at Yandex. We refer to it as to the *experimental* dataset. In order to simulate a real-life scenario, the datasets are sampled over consequent non-overlapping time periods, with the user modelling dataset preceding the experimental dataset.

Table 1: Experimental datasets statistics.

Name	#queries	#impressions	#CM sessions	winner
E_1	1,311	181,981	3,682,895	A
E_2	524	82,008	2,771,280	B
E_3	468	119,287	3,198,372	A
E_4	1,502	109,596	5,691,939	A
E_5	1,255	52,314	2,045,122	A
E_6	279	9,175	1,100,874	B

In order to collect the user modelling dataset, we apply the following filtering: firstly, we exclude all sessions that are affected by any online experiment, as that might result into a bias in the evaluation of the ability of the click model to predict the parameter $\boldsymbol{\mu}$; we remove all sessions with no documents clicked, as well as sessions with more than ten results examined since those can introduce an additional noise to the sDBN model. In order to balance the dataset size, the freshness of the click model parameters and the dataset sparseness, we use the following strategy: for the top 100 most frequent queries we collect the users' click behaviour over a period of week; for the rest of the queries we collected behaviour data from the eight previous weeks as well. All queries are normalised to lowercase. The same user modelling dataset is used in all experiments.

The interleaving experiments are sampled from the query log, starting from day after the click modelling dataset time span ended. The experimental data represents the users' click behaviour recorded while performing six ($E_1 ... E_6$) Team Draft-based interleaving experiments in April-May, 2013. In order to reduce variance in the offline evaluation, for a particular interleaving experiment, we keep only those queries that have every possible combination of an interleaved result page and a credit assignment rule (under the Team Draft algorithm) shown to the users at least once. Next, in order to avoid sparsity, we consider only the top six results in each result list. To further reduce noise, we exclude queries with results that are examined less than two times in the user modelling dataset. Queries with equal result lists for both A and B are removed as non-informative under the deduped credit assignment. The total number of user sessions used to train the click model parameters for queries in the experimental dataset is referred to as click modelling sessions (CM sessions), We present the statistics on the whole dataset in Table 1. In the experiments considered, the A ranking function represents the production search system, while B corresponds to the experimental ranking. Winners are defined as a result of Team Draft on the considered interleaving dataset (bootstrap test, $p \leq 0.05$).

Having introduced the dataset used in the experimental part of our paper, we proceed to discussing the evaluation methodology in the next section.

6. EVALUATION METHODOLOGY

Our experimental study has the following goals. The first goal is to ensure that solving the optimisation problem (10), stated in Section 3, indeed helps to maximise the interleaving sensitivity. The second goal is to study the effectiveness of the method proposed to estimate the parameter $\boldsymbol{\mu}$, described in Algorithm 2. Finally, we aim to investigate how our approach compares to the baseline interleaving method when several credit aggregation schemes are considered.

However, before addressing these experimental goals, we firstly discuss the metrics (Section 6.1) and statistical approaches (Section 6.2) used in our study. After that, we discuss the baseline (Section 6.3) and the credit aggregation schemes considered in our experiments (Section 6.4).

6.1 Metrics used

In order to measure how good the predicted values of $\boldsymbol{\mu}$ are, we use the following idea. Let us assume that we have a predicted value of the parameter $\boldsymbol{\mu}$, $\boldsymbol{\mu}_{predict}$ and $\boldsymbol{\pi}_{predict}$ is the corresponding optimal policy. In addition, the true value $\boldsymbol{\mu}_{true}$ can be directly calculated using Equation (8) once the full experiment dataset is available. The optimal policy corresponding to $\boldsymbol{\mu}_{true}$ is referred to as $\boldsymbol{\pi}_{true}$. From the literature we adapt the *regret* metric as a measure of optimality of $\boldsymbol{\pi}_{predict}$:

$$R = \boldsymbol{\mu}_{true}\boldsymbol{\pi}_{true} - \boldsymbol{\mu}_{true}\boldsymbol{\pi}_{predict} \tag{14}$$

The regret represents the loss in the value of objective function (10a) due to using the estimated value $\boldsymbol{\mu}_{predict}$ instead of the true value $\boldsymbol{\mu}_{true}$. In other words, the regret shows how good the solution of the optimisation problem (10) with the *predicted* parameters is in comparison with the solution of the problem with *ground-truth* parameters. Therefore if the parameters are predicted ideally, the corresponding solution has zero regret. In order to emphasise the relative importance of queries, we weight each query by the number of impressions and report the average values for each experiment.

In order to measure the sensitivity of an interleaving method, we leverage two approaches. Firstly, similar to the previous work of Radlinski and Craswell [13], and Chapelle et al. [3], we study the probabilities of obtaining the correct experiment outcome after bootstrapping k impressions from the existing experimental dataset. This quantity can also be considered as the amount of user impressions an algorithm has to observe to reach a fixed p-value. We report the results of these evaluations graphically with a pre-defined set of k. The number of bootstrap samples is set to 10,000.

Another measure used to estimate the interleaving sensitivity is *z-score*, as used in [3]. Denoting by $c(v_j)$ the aggregated credit assigned as a result of the impression v_j, we define a sample mean $\bar{\Delta}$ as $\bar{\Delta} = \frac{1}{N}\sum_j c(v_j)$. Furthermore, as $\bar{\Delta}$ is approximately normally distributed for large numbers of impressions N, we can calculate a method's confidence by z-score as follows:

$$z_c = \frac{\bar{\Delta}}{\sigma_c}\sqrt{N} \tag{15}$$

where σ_c is the standard deviation of $c(v_j)$. As noted by Chapelle et al. [3], if an interleaving algorithm has a z-score m times higher than another, which indicates that the latter needs m^2 more data to achieve the same confidence. Thus, the confidence of the algorithm in the experiment outcome is directly connected to its sensitivity: an algorithm with higher confidence is more sensitive.

6.2 Statistical methodology

As we aim to investigate how the required number of user sessions to achieve a particular level of confidence changes for the considered interleaving algorithms, the evaluation must be performed on the impression level. For this reason, the query-level methodology used by Radlinski and Craswell [13] to measure sensitivity is not directly applicable in our work.

```
Input: Dataset of user impressions $D$, sampled under
       policy $P_S$; policies $P_S$ and $P_T$; number of
       bootstrap samples $K$, size of a sample $S$
Output: Probability of error $P(e|S)$
// calculating importance weights
for $j = 1..|D|$ do
    $w_j \leftarrow \frac{P_T(v_j)}{P_S(v_j)}$
end
// normalise so $\boldsymbol{w}$ is a distribution
$\boldsymbol{w} \leftarrow \frac{\boldsymbol{w}}{\sum_j w_j}$
$a \leftarrow 0$ //agreement counter
for $k = 1..K$ do
    $s \leftarrow$ sample of $S$ impressions from $D$ with
    replacement, according distribution $w$
    if $\Delta(s) \cdot \Delta(D) > 0$ then
        $a \leftarrow a + 1$
    end
end
$P(e|S) = 1 - a/K$
```

Algorithm 3: Estimating the error probability $P(e|S)$ given a sample of size S.

Since our experiments leverage datasets that have been obtained from the Team Draft-based experiments, the evaluation of other algorithms is a challenging task. Indeed, we need to estimate the sensitivity of an interleaving algorithm once it is deployed, based on the data obtained from running another algorithm. This task is related to the *off-policy* evaluation and has previously been discussed in [8, 16]. Before discussing the method we used to approach this task, we introduce the required notation. Firstly, we assume that the available experiment dataset was generated from a source distribution P_S from one interleaving method and we want to evaluate some statistic (e.g. sensitivity) of another interleaving method which, once deployed, would generate data under the target distribution, P_T. In all our experiments the source distribution P_S corresponds to the distribution of the result pages, generated by the Team Draft algorithm.

In order to estimate the expectation of a statistic $f(v)$ of the dataset D of impressions v_j, the importance sampling estimator can be used [8, 16]:

$$\mathbb{E}(f) = \frac{1}{N}\sum_{j=1}^{N} f(v_j)\frac{P_T(v_j)}{P_S(v_j)} \tag{16}$$

The core of the importance sampling estimator is to reweight samples obtained under the source distribution, so that the expectation equates to the one of the target distribution. Since the probability of submitting a query by the users is independent of the interleaving algorithm deployed, only the probability of finding a particular combination of an interleaved results page and credit function in a query stream depends on the interleaving algorithm. Further, we assume that we are comparing two algorithms with the same space of generated interleaved result lists L. Let us denote the query submitted in impression v_j as q_j, and the result list demonstrated by L_j. Then the experiment outcome $\bar{\Delta}_T$ can be found using the following expression:

$$\bar{\Delta}_T = \frac{1}{N}\sum_{j=1}^{N} c(v_j)\frac{\pi_{T,j}}{\pi_{S,j}} \tag{17}$$

where $\pi_{T,j}$ and $\pi_{S,j}$ denote the probabilities of showing result list L_j according to the policy of the evaluated (target) and available (source) algorithms[4]. We apply the same

[4] It is also required for $\pi_{S,j}$ to be positive for all L_j occurring in the dataset, however this holds for the Team Draft policy.

Table 2: Learning the optimisation problem parameters regret. TD denotes Team Draft, the solution of (10) with the parameter $\boldsymbol{\mu}$ predicted by Algorithm 2 is denoted by UB. α is set to 0.5, which is selected according to Table 3.

	E_1	E_2	E_3	E_4	E_5	E_6
TD, $\times 10^{-2}$	1.80	1.23	1.20	0.88	0.48	3.36
UB, $\times 10^{-2}$	**1.53**	**1.12**	**1.13**	**0.82**	**0.45**	**3.23**

approach to estimate the variance of $c(v_j)$, σ_T. In turn, both $\bar{\Delta}_T$ and the variance σ_T are used to obtain the importance sampling estimate of the algorithm's z-score metric, discussed in Section 6.1.

Another approach to measure an interleaving algorithm's sensitivity that was discussed in Section 6.1 is to study the number of impressions required for the algorithm to define an experiment outcome with a particular certainty. Usually, bootstrapping is used to estimate the probability of obtaining the correct (ground-truth) experiment outcome provided a fixed number of user impressions from the experiment [13, 19]. However, the direct approach is not applicable in our case, since only the dataset generated under the Team Draft algorithm is available. In order to obtain these estimates, we perform a modified bootstrap sampling by means of Algorithm 3. Similarly to the importance sampling, Algorithm 3 re-weights the user impressions from the dataset. However these weights are further used as probabilities to sample from the dataset (after normalisation). This algorithm is related to the *sampling-importance-resampling* (SIR) [2] algorithm to obtain i.i.d. samples from the target distribution P_T once a dataset under source distribution P_S is provided. The algorithm is intuitive: if a point v_i appears in the dataset with probability $P_S(v_i)$, after sampling with probability proportional to $\frac{P_T(v_i)}{P_S(v_i)}$, it will be included into the sample with probability $P_T(v_i)$.

6.3 Baseline interleaving method

We believe it is interesting to compare our approach to the Optimised Interleaving approach proposed by Radlinski and Craswell [13]. However this is not feasible in our experimental setting. In contrast to [13], we base our evaluation on real-life Team Draft experiments and, as a result, the set of possible result lists L we can consider coincides with the one of Team Draft. Unfortunately, we found that for a considerable part of the dataset the optimisation problem (10) is infeasible if the result lists generated by Team Draft are combined with the credit assignment schemes from Optimised Interleaving.[5] Similarly, the availability of the user interaction data only for Team Draft-based result lists makes it impossible to compare the proposed approach with Probabilistic Interleaving method [9] directly. For these reasons, we consider Team Draft as the baseline in our experiments.

6.4 Credit aggregation schemes

Our proposed approach to improve the interleaving sensitivity can be used with various credit assignment and aggregation schemes. Note that our approach to increase the sensitivity of the algorithms and approaches relying on the modification of the credit function are not mutually exclusive, but, actually, complimentary to each other. In our experimental study, we investigate the sensitivity while applying two credit aggregation schemes [3], namely: *deduped binary* and *deduped click*. Both credit aggregation schemes are closely related to the algorithm of building an interleaved result list by Team Draft. Firstly, if the top $\bar{r}$ results for both A and B are equal, then $\delta_b(r)$ equates to zero for all positions above $\bar{r}$:

$$\bar{r} = max_r\{\forall j \ \leq rA(j) = B(j)\} \Rightarrow \forall r \leq \bar{r} \ \delta_b(r) = 0 \quad (18)$$

For the positions below $\bar{r}$, the credit for a click is assigned to the *team* (A or B) the result belongs to [14].

However, these schemes differ in the way credits are aggregated. In the case of the deduped binary scheme, for each impression v_j a single winner is selected and its score is incremented:

$$c_b(v_j) = sign(C_A - C_B)$$

In contrast, in the deduped click scheme, the full credit is contributed:

$$c_f(v_j) = C_A - C_B$$

We expect that advanced machine-learned click weighting functions, as introduced by Yue et al. [19], can be combined with the proposed approach and further benefit the interleaving sensitivity. However, as our goal is to demonstrate the utility of interleaving sensitivity optimisation by adjusting the experiment policy, we leave advanced click weighting functions for future work.

7. RESULTS AND DISCUSSION

To examine the quality of the prediction of the optimisation problem parameters, in Table 2 we report the regret values for the considered algorithms. It can be seen that the optimisation with predicted values of $\boldsymbol{\mu}$ indeed results into lower regret (i.e. higher values of the non-regularised objective function (10a)), in comparison with non-optimised uniform solution of the problem (10), which is represented by Team Draft. We conclude that the learning of parameters $\boldsymbol{\mu}$ by Algorithm 2 succeeds in providing reliable estimates of the user behaviour. Thus we can expect that the interleaving sensitivity might be improved with the proposed approach in comparison with Team Draft.

In order to test this expectation, we perform the sensitivity analysis and report the results in Table 3. We report the relative z-scores of Team Draft and the proposed algorithms for two credit aggregation schemes: deduped click and deduped binary. The z-scores are normalised so that the z-score of Team Draft with the deduped click credit function equals 1 for each experiment (each row). This allows an intuitive interpretation of the results: e.g. in E_3 our proposed algorithm with $\alpha = 0$ with the deduped binary click aggregation has a relative z-score of 2.02, thus Team Draft with deduped click credit needs $2.02^2 = 4.08$ times more data to become that confident (Section 6.2). In order to get additional insights in the performance of our proposed algorithm, we vary α in the pre-defined set $\alpha \in \{0, 2^{-5}, 2^{-4}, ..., 2^1\}$, where $\alpha = 0$ corresponds to the case with no regularisation, i.e. to the objective function (10a). Team Draft corresponds to $\alpha \to +\infty$. The z-scores are estimated by means of importance sampling, as discussed in Section 6.2.

In general, higher values of α indeed force the solution of (10) to be similar to Team Draft: the values in the row

[5] For instance, A=$\{d_1, d_2, ..., d_9, d_{10}\}$ and B=$\{d_1, ..., d_8, d_{10}, d_{11}\}$. Then there are two possible interleaved result lists with the following inverse rank credit functions [13]: $\delta_1 = (0, ..., 0, \frac{1}{9}, \frac{1}{10} - \frac{1}{9})$ and $\delta_2 = (0, ..., 0, -\frac{1}{9} + \frac{1}{10}, \frac{1}{9})$. With these credit assignment functions Eq. (10b) becomes infeasible.

Table 3: Comparison of the sensitivity of interleaving optimised with historical user behaviour for different values of the trade-off parameter α and click aggregation schemes. The sensitivity is measured by z-scores, normalised so that in each experiment the z-score of Team Draft with the deduped click credit aggregation scheme equals to 1. For each α and click aggregation scheme combination the median improvement over experiments $E_1...E_6$ are reported. TD stands for Team Draft.

Exp	Deduped click credit aggregation								Deduped binary credit aggregation								
	$\alpha = 0$	1/32	1/16	1/8	1/4	1/2	1	2	$\alpha = 0$	1/32	1/16	1/8	1/4	1/2	1	2	TD
E_1	1.01	**1.02**	1.00	0.96	0.96	0.93	0.95	0.97	1.13	1.12	**1.13**	1.12	1.08	1.05	1.07	1.09	1.11
E_2	0.70	0.71	0.85	0.97	0.97	1.04	**1.06**	1.03	0.72	0.69	0.72	0.87	1.01	1.09	**1.11**	1.08	1.06
E_3	**1.70**	1.53	1.40	1.33	1.33	1.20	1.10	1.05	**2.02**	1.76	1.57	1.43	1.36	1.24	1.14	1.09	1.05
E_4	**1.21**	1.14	1.09	1.07	1.07	1.05	1.02	1.01	**1.31**	1.13	1.15	1.09	1.07	1.04	1.01	1.01	1.00
E_5	**1.58**	1.42	1.27	1.15	1.15	1.06	1.02	1.01	**1.64**	1.60	1.46	1.30	1.18	1.08	1.03	1.01	0.99
E_6	1.56	**1.61**	1.33	1.27	1.27	1.18	1.09	1.04	1.96	1.99	**2.03**	1.68	1.55	1.40	1.27	1.22	1.16
median	1.39	1.28	1.18	1.11	1.11	1.06	1.04	1.02	1.48	1.37	1.31	1.21	1.13	1.09	1.09	1.09	1.06

corresponding to $\alpha = 2$ and deduped click credit aggregation scheme are close to 1. Similarly, in the case of deduped binary credit aggregation the rows corresponding to $\alpha = 2$ and to Team Draft have close values. On the contrary, low values of α allow the optimised policy to diverge from Team Draft, so that the interleaving becomes "risky" and relies more on the noisy predictions of the user behaviour. As a result, in three experiments (E_3, E_4, E_5), the best result is achieved with $\alpha = 0$ and in two experiments (E_1, E_6) the best performance is achieved with $\alpha = 1/16$. Moreover, with $\alpha = 0$ the maximal median improvements of 1.39 and 1.48 are achieved for the deduped click and deduped binary aggregation schemes, respectively. These values correspond to approximately 48% reduction in the number of impressions required in comparison with Team Draft with the same credit aggregation scheme, to achieve the same level of confidence. However, there are downsides for behaving too risky: the sensitivity in experiment E_2 falls to 0.70 with $\alpha = 0$ and a deduped click credit aggregation. In turn, for values of α above 1/2, the sensitivity of the proposed algorithm outperforms that of Team Draft under both credit aggregation schemes in E_2. This supports the role of α as a parameter that trades off the level of risk (i.e. variance) due to noisy user feedback and the level of possible improvement. Moreover, these observations also suggest that with a proper tuning of α, it is possible to achieve an appropriate trade-off between the median improvement and the largest descrease in the interleaving sensitivity. On analysing Table 3, we note that the median sensitivity values of the proposed approach are above 1 for the deduped click credit aggregation and above the results of Team Draft for the deduped binary credit aggregation, demonstrating the advantage of our approach.

Another observation that can be made from Table 3 is that in five out of six experiments, Team Draft with the deduped binary click aggregation has relative scores that are not less than one. This indicates that Team Draft with the deduped binary credit aggregation has higher sensitivity than with the deduped click aggregation. This fact is inline with the results obtained by Chapelle et al. [3]. A possible explanation is that aggregating credit on the impression level reduces noise and thus increases the sensitivity. The same observation holds for our proposed algorithm: for all values of α and all of the experiments considered, the deduped binary scheme outperforms the deduped click aggregation, except for experiment E_2 and $\alpha = 1/32$. This observation also supports our decision to rely on the non-binary click aggregation scheme while deriving the optimisation problem (10) in Section 3, since the obtained solution can still be improved by the impression-level aggregation of credits.

A visual representation of these results is presented in Figure 1. The plots correspond to experiments from E_1 to E_6, and each plot represents the probability of obtaining an incorrect experiment outcome after considering a particular number of impressions, as calculated by Algorithm 3. We present results that correspond to Team Draft (i.e. $\alpha \to \infty$) and the solution of the optimisation problem (10) with $\alpha \in \{0.0, 0.5\}$[6] and the deduped credit aggregation scheme used. We notice that the results are consistent with those in Table 3: the proposed algorithm outperforms Team Draft in most the of experiments, and by varying α it is possible to control the optimisation risk: with $\alpha = 0$ the proposed algorithm demonstrates high sensitivity gains in E_3, E_4, E_5, E_6 but underperforms in E_2 with respect to Team Draft. On the contrary, with $\alpha = 0.5$ the maximum loss is almost negligible (E_1) while there are still significant improvements in E_3, E_4, E_5, E_6 and a slight improvement in E_2.

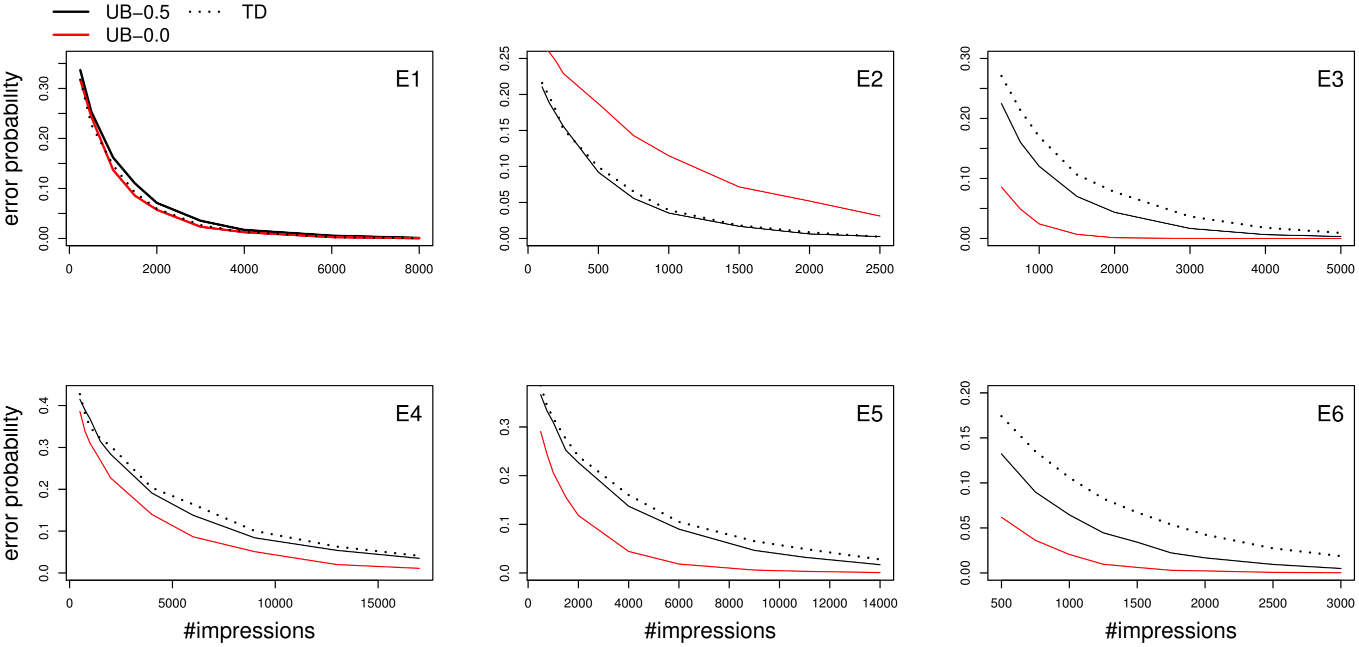
Overall, the results obtained suggest that the historical user behaviour information can be used to improve the sensitivity of the interleaving algorithms. The proposed interleaving algorithm, which adjusts the interleaving policy according to the solution of the optimisation problem (10), has generally a higher sensitivity than Team Draft, reaching up to a median of 48% decrease in the required number of impressions to achieve the same level of confidence. Moreover, the algorithm's regularisation parameter α can be used to select the appropriate variance in the sensitivity gains.

8. CONCLUSIONS AND FUTURE WORK

In this paper, we studied the problem of improving the interleaving sensitivity. We proposed a novel theoretically-motivated approach to optimise the interleaving experiment parameters using the historical user behaviour data routinely collected by search engines. Informally, the approach aims to predict what interleaved result pages are likely to contribute little to the experiment outcome and, after that, show them to the users as rarely as possible. In order to achieve this goal, a click model is trained on the historical click data. In turn, this model is used to predict future user behaviour on the interleaved result pages (that were possibly never shown to the users before). Next, we proposed to ad-

[6] Since $\alpha = 0.5$ is an appropriate trade-off among the values considered in Table 3 and $\alpha = 0$ is an interesting case of the non-regularised version of the optimisation problem (10).

Figure 1: Comparison of the interleaving methods sensitivity. TD denotes Team Draft, UB-0.0 and UB-0.5 correspond to interleaving optimised with respect to historical user behaviour with α equal to 0.0 and 0.5, respectively. The deduped binary credit aggregation scheme is considered.



just the probabilities of showing interleaved result pages so that a predicted upper bound on the experiment outcome is maximised. Further, since the predicted user behaviour can be noisy, we introduced a regularisation trade-off parameter which can be used to adjust the level of risk.

In order to test the proposed approach empirically, we performed an offline empirical study, which leverages data from the Team Draft interleaving experiments previously performed by a commercial search engine. In our study, we used the data obtained over six interleaving experiments, representing 555K user impressions in total. We investigated the sensitivity of the proposed approach with the previously proposed deduped click and deduped binary credit aggregation schemes. Our results suggest that with an appropriate tuning of the trade-off parameter, the proposed approach outperforms the Team Draft algorithm in ensuring the correct experiment outcome with up to the median of 48% less impressions.

An interesting direction for future work is to study a per-query strategy for selecting the risk parameter α. Indeed, the optimisation can be performed more aggressively for queries with a sufficient amount of historical click data available. Finally, we believe it is promising to further investigate other combinations of the credit assignment schemes and the proposed approach.

9. REFERENCES

[1] P. N. Bennett, F. Radlinski, R. W. White, and E. Yilmaz. Inferring and using location metadata to personalize web search. *SIGIR '11*.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[3] O. Chapelle, T. Joachims, F. Radlinski, and Y. Yue. Large-scale validation and analysis of interleaved search evaluation. *Transactions on Information Systems*, 30(1), 2012.

[4] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. *WWW '09*.

[5] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. *WSDM '08*.

[6] L. A. Granka, T. Joachims, and G. Gay. Eye-tracking analysis of user behavior in www search. *SIGIR '04*.

[7] F. Guo, C. Liu, A. Kannan, T. Minka, M. Taylor, Y.-M. Wang, and C. Faloutsos. Click chain model in web search. *WWW '09*.

[8] K. Hofmann, S. Whiteson, and M. de Rijke. Estimating interleaved comparison outcomes from historical click data. *CIKM '12*.

[9] K. Hofmann, S. Whiteson, and M. de Rijke. A probabilistic method for inferring preferences from clicks. *CIKM '11*.

[10] T. Joachims. Optimizing search engines using clickthrough data. *KDD '02*.

[11] T. Joachims. Evaluating retrieval performance using clickthrough data. In J. Franke, G. Nakhaeizadeh, and I. Renz, editors, *Text Mining*. Physica/Springer Verlag, 2003.

[12] F. Radlinski and N. Craswell. Comparing the sensitivity of information retrieval metrics. *SIGIR '10*.

[13] F. Radlinski and N. Craswell. Optimized interleaving for online retrieval evaluation. *WSDM '13*.

[14] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? *CIKM '08*.

[15] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. *WWW '07*.

[16] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[17] D. Tang, A. Agarwal, D. O'Brien, and M. Meyer. Overlapping experiment infrastructure: more, better, faster experimentation. *KDD '10*.

[18] E. M. Voorhees. The philosophy of information retrieval evaluation. In *Evaluation of Cross-Language Information Retrieval Systems*, LNCS, pages 355–370. 2002.

[19] Y. Yue, Y. Gao, O. Chapelle, Y. Zhang, and T. Joachims. Learning more powerful test statistics for click-based retrieval evaluation. *SIGIR '10*.