Generalized Team Draft Interleaving

Eugene Kharitonov[1,2], Craig Macdonald[2], Pavel Serdyukov[1], Iadh Ounis[2]
[1]Yandex, Russia
[2]University of Glasgow, UK
[1]{kharitonov, pavser}@yandex-team.ru
[2]{craig.macdonald, iadh.ounis}@glasgow.ac.uk

ABSTRACT

Interleaving is an online evaluation method that compares two ranking functions by mixing their results and interpreting the users' click feedback. An important property of an interleaving method is its sensitivity, i.e. the ability to obtain reliable comparison outcomes with few user interactions. Several methods have been proposed so far to improve interleaving sensitivity, which can be roughly divided into two areas: (a) methods that optimize the credit assignment function (how the click feedback is interpreted), and (b) methods that achieve higher sensitivity by controlling the interleaving policy (how often a particular interleaved result page is shown).

In this paper, we propose an interleaving framework that generalizes the previously studied interleaving methods in two aspects. First, it achieves a higher sensitivity by performing a joint data-driven optimization of the credit assignment function and the interleaving policy. Second, we formulate the framework to be general w.r.t. the search domain where the interleaving experiment is deployed, so that it can be applied in domains with grid-based presentation, such as image search. In order to simplify the optimization, we additionally introduce a stratified estimate of the experiment outcome. This stratification is also useful on its own, as it reduces the variance of the outcome and thus increases the interleaving sensitivity.

We perform an extensive experimental study using large-scale document and image search datasets obtained from a commercial search engine. The experiments show that our proposed framework achieves marked improvements in sensitivity over effective baselines on both datasets.

Categories and Subject Descriptors: H.3.3 [Information Storage & Retrieval]: Information Search & Retrieval
Keywords: interleaving; online evaluation

1. INTRODUCTION

Online evaluation approaches, such A/B testing and interleaving, are crucial tools in modern search engine evaluation [2, 5, 7, 11, 12, 13]. These approaches leverage the implicit feedback of the real users to evaluate changes to the search engine and can be applied even when the offline evaluation approaches might be impractical [19].

Since the online evaluation approaches rely on the noisy user feedback, a considerable number of observations is needed before a statistically significant conclusion can be made [2, 12]. Usually, an A/B testing experiment is deployed for a week or two [12]. A typical length of an interleaving experiment used in the literature is up to five days [2]. Such a long duration of the online experiments considerably limits their usefulness, and bounds the rate of the search engine's evolution.

Another concern is that a considerable fraction of the changes evaluated online turn out to actually degrade the user's search experience [12]. When evaluated in an online experiment during a period of a week, such a change increases the users' frustration with the results and might even force them to switch to another search engine.

These observations support the need to increase the speed of online evaluation experiments. When comparing two web document search ranking functions, interleaving is faster to obtain the comparison outcome than an A/B test [2, Section 7]. A variety of methods were proposed to further reduce the duration of interleaving experiments by improving the interleaving sensitivity. Roughly, this research can be divided into two areas: optimization of the credit assignment [2, 14, 20]; and optimization of the probability of showing of the interleaved result pages (interleaving policy) [10, 15]. In both areas, only document search has been studied so far. Furthermore, the current research in the interleaving policy optimization explicitly relies on a user model that is specific to the list-based representation.

In this paper we propose an interleaving framework that generalizes the existing research in two aspects. First, we consider both the interleaving policy and the credit assignment function as optimized parameters in our framework. As a result, our framework has a higher flexibility that can be used to achieve a higher sensitivity. Second, we formulate our framework to be general w.r.t. the actual presentation of the result pages, so that it can be applied for the domains such as image search, where grid-based presentation is used.

In order to simplify the parameter optimization procedure, we propose to use a stratified estimate of the interleaving experiment outcome, where the stratification is performed according to the teams of the results on the result pages shown. We demonstrate that our proposed stratification approach is also useful on its own, as in some cases it considerably increases the interleaving sensitivity.

Overall, the contributions of our work are three-fold:

- We propose a principled, data-driven framework to develop sensitive interleaving that combines the stratification, the interleaving policy optimization, and the

CIKM'15, October 19–23, 2015, Melbourne, Australia.
© 2015 ACM. ISBN 978-1-4503-3794-6/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2806416.2806477.

credit function learning in a single framework that can be applied in domains with the list-based and the grid-based result presentations;

- We propose sufficient conditions that the click feature representation and the interleaving policy need to satisfy so that the resulting interleaving method remains unbiased;
- We perform a large-scale evaluation study of the proposed framework, using two datasets that contain document and image search online experiments.

The remainder of this paper is organised as follows. In Section 2 we discuss the related work. In Section 3 we define our interleaving framework and discuss its details in Section 4. Our proposed stratification technique, and how to optimize the interleaving parameters, is discussed in Sections 5 and 6, respectively. In Section 8 we describe the instantiations of our framework for the web document search and for the image search domains. The datasets and the evaluation scenario we use are described in Section 7 and 9, respectively. We discuss our obtained results in Section 10. We conclude this paper and discuss future work in Section 11.

2. RELATED WORK

Since the introduction of the first interleaving method, Balanced Interleaving [7, 8], several other interleaving methods were proposed, including Team Draft [16], Probabilistic Interleaving [6], Optimized Interleaving [15]. An important characteristic of an interleaving method is its sensitivity, i.e. ability to obtain a reliable experiment outcome with as few user interactions as possible. The problem of increasing the sensitivity of an interleaving method has attracted a considerable attention from the research community, and below we review the most relevant work in this area.

Yue et al. [20] proposed a method to learn a more sensitive credit assignment function for the Team Draft interleaving experiments. Later, this approach was also discussed by Chapelle et al. [2]. Informally, the core idea of [20] is to learn how to weight user clicks in the interleaving comparisons so that the confidence in the already performed experiments is maximized. As a result, new interleaving experiments will achieve the required level of confidence in their outcomes with fewer user interactions, i.e. the interleaving method will have a higher sensitivity. Yue et al. refer to this learning problem to as an "inverse" hypothesis test: given user interaction data for the comparisons with known outcomes, one learns a credit assignment function that maximizes the power of the test statistic in these comparisons.

Our work is based on the ideas of Yue et al. [20], and aims to overcome some of the shortcomings of their approach. First, it is not straightforwardly clear what kind of features and weighting functions are allowed so that no biases are introduced when learning the credit assignment function. It is possible to build an example of the click feature representation that make the credit function learning process prone to biases (Section 4). In our work, we propose a formal unbiasedness requirement that ensures that a feature-based credit assignment function is not biased. Moreover, we propose a restricted family of the click features that allow us to make this requirement easy to operate in practice.

In their work, Yue et al. assume that the interleaving policy (the probabilities of showing of the different interleaved result pages) is fixed. We propose to optimize both the interleaving policy and the credit assignment function jointly, and this results in a higher interleaving sensitivity.

Radlinski and Craswell [15] proposed the Optimized Interleaving framework, which specifies a set of requirements that an interleaving method has to meet so that (a) its results are not biased, (b) it is sensitive, and (c) the users are not too frustrated by the interleaved results pages. Our framework is based on Optimized Interleaving, but it also has significant differences from it. First, Optimized Interleaving is formulated specifically with a particular web document search user model in mind, and the interleaving policy optimization it performs is formulated with respect to a click model that is specific for the list-based result presentation. This hinders extending the interleaving approaches to other domains. In contrast, we propose a generalized unbiasedness requirement, that can be applied for the grid-based result pages. Second, we perform a joint data-driven optimization of the interleaving policy and the credit assignment function. In contrast, in Optimized Interleaving, the credit assignment function is fixed; and the interleaving policy is optimized with respect to a randomly clicking user, in a data-free manner. Moreover, the interleaving policy optimization considered by Radlinski and Craswell is performed on a per-query basis. This makes the evaluation runtime system more sophisticated, as this optimization must be performed each time a long-tail query is submitted. Finally, due to using a large set of possible interleaved result pages that is different from the result pages generated by Team Draft and Balanced Interleaving, it is hard to perform a representative evaluation study of this method without actually implementing it and deploying a large set of real-life experiments. Unlike Optimized Interleaving, our proposed framework has the same interleaving policy for all queries and experiments which is fixed once learned, thus it imposes little implementation costs over the standard Team Draft interleaving. Further, its performance can be evaluated using a set of historical experiments, available to each search engine that uses Team Draft-based interleaving experiments.

While studying a different problem of the document search multileaving, Schuth et al. [18] used the variance of the outcome as a proxy for the interleaving sensitivity. This approach can be considered as a hybrid between approaches in [20] and [15]: the optimization is performed w.r.t. to a randomly clicking user, as in [15]; the optimization objective is close to z-score used in [20], as the latter favours a lower variance, too. We believe this approach can be suboptimal in comparison to the data-driven optimization used by Yue et al. [20] and our framework. Indeed, in the case of the sufficiently large dataset, the parameters can be optimized based on the real-life data, without relying on a model of a randomly clicking user.

Kharitonov et al. [10] addressed the problem of improving the interleaving sensitivity by predicting the future user clicks using the historical click data. Their approach explicitly relies on the document search click models, thus it is hard to generalize to other domains. Moreover, [10] requires the search engine to store per-query click models in runtime, and use them when interleaving the result pages. It is not clear how practical this requirement can be in the presence of the long-tail queries that form a considerable part of the query stream. Further, as it can be applied only for the head queries with click data available, it is not clear if any additional biases can occur due to increased sensitivity to the changes in the top queries. In contrast, our framework does not requires significant changes in the search engine's experimentation infrastructure and can be applied for the domains with the grid-based presentation of the results.

In their work [3], Chuklin et al. proposed an interleaving method that goes beyond the classic "ten blue links" web document presentation and deals with the vertical results (e.g. News, Images, Finance) incorporated in the main web search result page. However, the challenges that Chuklin et al. address (e.g., ensuring that in the interleaved result page vertical results are still grouped) are quite different from the problems faced when developing an interleaving mechanism for a new domain. In the latter case, one needs to decide how to specify the credit assignment function, how to select the interleaving policy, etc. To the best of our knowledge, our work is the first to address the problem of interleaving in a domain with the grid-based result presentation.

One of the approaches to improve the interleaving sensitivity we discuss is stratification, a simple yet effective technique that has its roots in the Monte-Carlo stratified sampling methods [1, 17]. Previously, its application for online A/B tests was studied by Deng et al. [4], but it was never considered in the context of interleaving.

Overall, our framework finds a solid foundation in the research discussed above, but it also addresses several shortcomings of the earlier approaches. In the next section we formally introduce it.

3. FRAMEWORK DEFINITION

First, we informally outline how interleaving experiments are performed. Suppose, that we need to compare a changed system B to the production system A using an interleaving experiment. To do that, a random subset of the users is selected to take part in the experiment. When a query is submitted, both results from A and B are retrieved. Further, the interleaving policy is used to determine which of the possible mixed (interleaved) result pages to show to the user. Next, the users' clicks on the interleaved result page are observed, and the credit assignment function is used to infer the credits of the alternatives. After the experiment is stopped, the aggregated credits of the alternatives are compared. If B has a statistically significantly higher credit, it is accepted that B outperformed A.

The works of Yue et al. [20] and Radlinski and Craswell [15] lay the foundation for our framework. However, our framework has significant differences from [20] and [15]. Specifically, our proposed framework performs a joint optimization of the interleaving policy and the credit assignment function, while Yue et al. and Radlinski and Craswell optimize only one of these parameters. Further, our framework can be applied for search domains with grid-based result pages. Below, we provide a formal requirement that a feature-based credit assignment function, the click feature representation, and the interleaving policy have to meet for the interleaving to be unbiased. In contrast, Yue et al. do not discuss possible biases that can emerge due to feature-based learning, and Radlinski and Craswell only discuss simple, feature-less credit assignment rules. By addressing the above discussed gaps, we build a sensitive interleaving framework that generalizes approaches proposed by Yue et al. [20] and Radlinski and Craswell [15].

In our framework, we consider the result pages that are obtained by applying the Team Draft mixing algorithm [16] to the lists of the results of the underlying rankers A and B, sorted according to their relevance. The exact mapping of the sorted result list into a result page is domain-specific (the list-based for document search, or the grid-based for image search). Assuming that under this mapping the results ranked higher in the ranked list are mapped into positions with higher examination probability, mixing the sorted result lists of the rankers A and B according to Team Draft will result in a result page that cannot be more frustrating for the users than both the result pages generated from outputs of A and B. Due to this assumption we avoid the necessity of specifying the mixing algorithm for each possible domain-specific presentation, and can work with the underlying ranker output, which is always list-wise in practice. Apart from that, relying on the Team Draft-based result pages allows us to re-use a large-scale dataset of the experiments collected by a search engine for our evaluation study (Section 10).

The Team Draft mixing algorithm builds the interleaved result list in steps. At each step, both teams contribute one result each to the combined list. Each team contributes the result that it ranks highest among those that are not in the combined list. However, the team that contributes first at each step is decided by a coin toss. For instance, as there are usually 10 results on a document search result page, 5 coin tosses are required to build it. Thus, there are exactly $2^5 = 32$ different distributions of the result teams on a result page[1].

Now we can define the first component of our framework:

F1. The set $\{(L_i, T_i)\}_{i=1}^{l}$ of the pairs of the interleaved result pages $L_i \in \mathbb{L}$ and their corresponding distributions of the result teams $T_i \in \mathbb{T}$. The result pages $\mathbb{L}$ are obtained by applying the Team Draft [16] mixing algorithm to the sorted outputs of the rankers A and B, and further domain-specific presentation of the ranked list. We define $T_i(p)$ to be equal to 1 (-1) if the team of the result on position p of the interleaved list that produced L_i is B (A);

It is possible that some pairs (L_i, T_i) contain identical result pages L_i, despite that the team distributions T_i associated with them are different (e.g. if A and B produce identical result lists). We consider such pairs to be different.

Further, following [15] we explicitly define the interleaving policy as a parameter of the framework:

F2. An interleaving policy π, $\pi \in \mathbb{R}^l$ determines the probability of using a particular team distribution when building an interleaved result page: $\pi_i = P(T_i)$;

Under our framework, the interleaving policy is the same for all queries and interleaving experiments. Informally, it can be considered as a distribution over the random seeds that can be used to "initialize" the coin used in Team Draft.

From [20] we adopt the feature representation of the user's click $\phi(\cdot)$ and the form of the credit assignment function S:

F3. A function $\phi(\cdot)$ that maps a user click c on an interleaved result page to its feature representation $\phi(c) \in \mathbb{R}^n$. We also define an auxiliary indicator $T(c)$ that equates to 1 (-1) if the team of the clicked result is B (A);

F4. A scoring rule, $S = S(q; w) = \sum_{c \in q} T(c) \cdot w^T \phi(c)$ that maps a sequence of clicks in the interaction q to the score of the alternative B. The vector w is a parameter, $w \in \mathbb{R}^n$.

After running an experiment e, the score statistic $\Delta(e)$ can be calculated:

$$\Delta(e) = \frac{1}{|Q|} \sum_{q \in Q} S(q; w) \tag{1}$$

[1] They can be enumerated as *ababababab*, *ababababba*, *ababab-baab*, ..., *bababababa*.

where Q is a set of the user interactions in the experiment e. If $\Delta(e)$ is statistically significantly above zero, it is concluded that B outperforms A in the experiment e.

To ensure that the interleaving is unbiased, Radlinski and Craswell [15] suggested the following criterion for the document search scenario: a randomly clicking user should not create any preference between A and B. To formalize this idea, they considered a user who (a) samples the number of the considered top results k randomly and (b) clicks uniformly at random on η results from the top-k results. This formulation explicitly relies on a list-based presentation. Furthermore, in our case the formalization is even more challenging as the credit $S(q; w)$ is a function itself, since some feature representations might be prone to biases (we discuss this further in Section 4). We propose the following generalization of unbiasedness criterion from [15]:

R1. For any fixed sequence of clicks, the expectation of the total credit over the all pairs (L_i, T_i) of the interleaved pages L_i and distributions of teams T_i should be zero. Denoting the length of the sequence as J, the positions clicked as $p_1, p_2, ..., p_J$, and their corresponding click features as $\phi_1, \phi_2, ..., \phi_J$ we formalize this requirement as follows:

$$\forall J,\ \forall\{(p_j, \phi_j)\}_{j=1}^{J} \quad \sum_i \pi_i \cdot \sum_j T_i(p_j) \cdot w^T \phi_j = 0$$

Due to the linearity of the expectation, R1 is sufficient to guarantee the absence of the preferences for any randomized combination of the click sequences, too. Informally, this guarantees that a user who specifies an arbitrary interaction scenario that does not depend on the presented documents (e.g., "click on the first position, sample the dwell time uniformly from [0, 30], click on the third result, ...") will not create any preference for A or B in expectation.

Next, we require the policy π to be a valid distribution:

R2. $\forall i \quad \pi_i \geq 0; \quad \sum_i \pi_i = 1$

Among all of the possible combinations of $\{\pi, w\}$ that satisfy R1 and R2, we want to select the combination that maximizes the interleaving sensitivity. Based on [20], we use a *dissimilarity* measure D between compared alternatives in a set of historical experiments E as a proxy for the sensitivity in future experiments. Indeed, the more dissimilar the alternatives are, the easier it is to differentiate them.

O1. The optimal combination of parameters π and w should maximize the dissimilarity D over a set of experiments E:

$$\hat{\pi}, \hat{w} = \arg\max_{\pi, w} D(E, \pi, w)$$

This ends the framework description. In the next section, we discuss the requirement R1 in more detail.

4. UNBIASEDNESS REQUIREMENT

The motivation behind R1 is to ensure that a user who clicks according to a fixed pattern that does not depend on the results shown would not provide any preference for A or B. Clearly, if R1 is not satisfied, a certain bias towards one of the alternatives might appear.

To illustrate how such a bias might arise, let us consider the following "toy" example. Let us assume that the feature representation vector $\phi(c)$ is a two dimensional vector, with its first component $\phi_0(c)$ being equal to 1 if the clicked result is from A, and zero otherwise. Similarly, $\phi_1(c)$ is equal to 1 if click c is performed on a result from B. Suppose we fix the interleaving policy to be uniform, and learn the vector of weights w based on the dataset of experiments. It is possible that, as a result of the learning, the weights of the features will obtain different values, e.g. if the learning dataset has more experiments with A winning. This results in poor generalization capabilities and biased interleaving. By considering a user who always clicks on the first position, we notice that in our toy example R1 requires w_1 to be equal to w_2.

In this work we simplify R1 by using a restricted family of features. Namely, we use click features that do not depend on the result page[2] L_i. By restricting the set of possible features, we achieve an intuitive *symmetry* property: after swapping A and B ("renaming" A to B, and B to A), the experiment outcome $\Delta(e)$ will only change its sign, but not its absolute value (which is violated in our toy example). Furthermore, the following Lemma 1 shows the conditions that are sufficient to satisfy R1 if we restrict the used features:

LEMMA 1. *For a feature representation ϕ, and a policy π to satisfy R1, it is sufficient that:*

- *ϕ is independent from L_i;*
- *For each position p on the result page, the probability of observing a result from A must be equal to the probability of observing a result from B: $\forall p \quad \sum_i \pi_i \cdot T_i(p) = 0$.*

PROOF. First, using the independence of ϕ from L_i, we re-write R1 as follows:

$$\sum_j w^T \phi_j \cdot \sum_i \pi_i \cdot T_i(p_j) = 0 \qquad (2)$$

An obvious way to satisfy Equation (2) is to select π such that for any click position the expectation of T_i is zero for every position:

$$\forall p \ \sum_i \pi_i \cdot T_i(p) = 0 \qquad (3)$$

□

Lemma 1 provides us with a convenient approach to satisfy R1 while optimizing the interleaving parameters. Indeed, once we use only the features that are independent from the particular interleaved result pages shown, whether R1 is satisfied or not depends only on the interleaving policy. In that case, R1 reduces to the following equality constraint:

$$R\pi = 0 \qquad (4)$$

where $R \in \mathbb{R}^{m \times l}$ is a matrix with its element R_{ji} equal to the team $T_i(j)$ (1 or -1) of the result shown on jth position of the interleaved result page L_i.

Equation (4) gives an intuition how the optimization of the interleaving policy can be performed: the number of independent[3] equality constraints grows linearly as $m/2$ with the number of positions m, but the number of different team

[2] The features cannot depend on the clicked result, its team, and its position in A and B. In contrast, the features can depend on the properties of the clicks itself (e.g. the position of the click, its dwell time) and the total number of clicks.

[3] As discussed in F1, our framework relies on the Team Draft mixing algorithm. Due to its specifics, if Equation (3) holds for a position $2k$ and a policy π, it also holds for the position $2k + 1$ and π.

distributions $\mathbb{T}$ and thus the dimensionality of the policy vector π grows exponentially as $2^{m/2}$. As a result, some "degrees of freedom" appear that can be used to find a sensitive yet unbiased policy. This intuition is similar to the one behind the optimization in Optimized Interleaving [15].

5. STRATIFIED SCORING

In Section 3, the experiment outcome is calculated as a sample mean of the scores of the individual interactions $\Delta(e)$, Equation (1). This approach is similar to the one used previously [2, 8, 15, 16]. We propose to use a stratified estimate $\Delta_s(e)$, where the stratification is performed according to the distribution of the teams (*ababababab*, ...) on the result pages shown to the users. Further, by Q_i we denote the set of the user interactions where the distribution of the teams on the result page shown is T_i. Using this notation, our proposed stratified estimate can be estimated as follows:

$$\Delta_s(e) = \sum_i \pi_i \cdot \frac{1}{|Q_i|} \sum_{q \in Q_i} S(q; w) \tag{5}$$

Both the stratified estimate $\Delta_s(e)$ and the sample mean $\Delta(e)$ have the same expected values, but the variance of $\Delta_s(e)$ can be lower and, consequently, it has higher sensitivity. Indeed, denoting the number of interactions in the experiment e as N, the variance and the expectation of the interaction score S among the sessions in the ith stratum as $var_i[S]$ and $\mathbb{E}_i[S]$, and applying the law of total variance, we obtain:

$$\begin{aligned} var\left[\Delta(e)\right] &= \frac{\sum_i \pi_i \cdot var_i[S] + \sum_i \pi_i (\mathbb{E}_i[S] - \sum_i \pi_i \cdot \mathbb{E}_i[S])^2}{N} \\ &\geq \frac{1}{N} \sum_i \pi_i \cdot var_i[S] = var\left[\Delta_s(e)\right] \end{aligned} \tag{6}$$

Since the frequency of T_i is determined by π_i, the probability of each stratum is known and fixed before starting an interleaving experiment.

As can be seen from Equation (6), the stratification reduces the variance only when the inner-strata means $\mathbb{E}_i[S]$ are different from the overall mean $\sum_i \pi_i \cdot \mathbb{E}_i[S]$. In our proposed approach of Equation (5), the stratification is performed according to the teams of the results on a result page T_i. In the case of the document search, T_i is a strong indicator of the outcome of a single comparison, as it specifies, for instance, if the click on the first result is counted in favour of A or B.

The stratification alone can considerably improve the sensitivity of the interleaving experiments in some cases (Section 10). Moreover, as we discuss in Section 6, the use of the stratified outcome Δ_e considerably simplifies the optimization of the interleaving parameters.

6. OPTIMIZATION OF THE PARAMETERS

To specify an instantiation of our proposed interleaving framework, we need to specify the interleaving policy π, the feature representation $\phi(c)$, and the vector of weights w. The feature representation is domain-specific. However, our proposed approach to determine the vector of weights w and the interleaving policy π are the same irrespective of the domain. We adopt a data-centric approach [20] to select π and w and select them maximize the sensitivity on the previously collected data.

We assume that a dataset E of interleaving experiments is available, so that for each experiment in this dataset the user interactions are recorded, and the experiment outcome is known. Such a dataset can be obtained from running interleaving experiments by a search engine (e.g., Team Draft-based experiments) and selecting the experiments with a high confidence in the outcome [2, 20] or by deploying "data collection" experiments where B is obtained by manually degrading A, and all possible combinations of the result lists and the team distributions are shown to the users with the uniform policy. We discuss these two approaches in more detail in Section 8.

To simplify the notation, without any loss in generality, we further assume that in all experiments $e \in E$ the alternative B outperformed A so that $\Delta_s(e)$ is positive. If it is not the case in a particular experiment, A and B can be swapped for that experiment.

As stated in the sensitivity optimization objective $O1$, we want to find the values of parameters π and w that maximize the dissimilarity between A and B over the available experiments and satisfy constraints $R1$ and $R2$. Since the sensitivity of the interleaving does not depend on the scaling of w, to make the optimization problem well-posed, we additionally constrain w to have the unit norm. Overall, this results in a general optimization problem of the following form:

$$\hat{\pi}, \hat{w} = \arg\max_{\pi, w} D(E, \pi, w) \quad s.t.\ R1, R2,\ w^T w = 1$$

Further, we discuss two ways to specify the idea of dissimilarity, proposed by Yue et al. [20]: the *mean score* and the *z-score* dissimilarities.

Mean score We start with the simplest case, when dissimilarity is calculated as the mean value of the stratified score:

$$D_m(E, \pi, S) = \frac{1}{|E|} \sum_{e \in E} \sum_i \pi_i \frac{1}{|Q_{e,i}|} \sum_{c \in q, q \in Q_{e,i}} T(c) \cdot w^T \phi(c) \tag{7}$$

where $Q_{e,i}$ is the set of user interactions with the team distribution T_i demonstrated.

Further, we introduce a matrix X with its columns corresponding to the individual features, and rows corresponding to the strata, so that the element X_{kr} is equal to the mean value of the rth feature ϕ_r in the kth stratum:

$$X_{kr} = \frac{1}{|E|} \sum_{e \in E} \frac{1}{|Q_{e,k}|} \sum_{c \in q, q \in Q_{e,k}} T(c) \cdot \phi_r(c)$$

Using the introduced notation, the optimization objective can be re-written as follows:

$$D_m(E, \pi, w) = \pi^T X w$$

Thus, we are looking for π, w that maximize (8):

$$\begin{aligned} &\hat{\pi}, \hat{w} = \arg\max\nolimits_{\pi, w} \left[\pi^T X w\right] \\ &s.t.\ R1, R2,\ w^T w = 1 \end{aligned} \tag{8}$$

Finally, we notice that if we set π to be the uniform policy, the solution of the optimization problem (8) becomes similar to the solution of the corresponding case in Yue et al. [20]: w lies on the unit sphere $w^T w = 1$ and maximizes the dot product $\pi^T X \cdot w$, so $\hat{w} = \frac{\pi^T X}{||\pi^T X||_2}$. The difference is in the way X is calculated, as the scores are stratified in our case.

Z-score The second way to specify the level of dissimilarity between A and B proposed by Yue et al. [20] is to measure the z-score statistic. Informally, this measures how the distance between A and B is far from zero in terms of the variance of this distance.

Following Yue et al., we simplify the optimization by combining the set of experiments E into a single artificial experiment $\bar{e}$. In that case, the z-score can be calculated as follows:

$$D_z(E, \pi, w) = \frac{\Delta_s(\bar{e})}{\sqrt{var\left[\Delta_s(\bar{e})\right]}} \tag{9}$$

As earlier, we introduce a matrix X with its elements equal to the per-stratum means of the individual features:

$$X_{kr} = \frac{1}{|Q_{\bar{e},k}|} \sum_{c \in q, q \in Q_{\bar{e},k}} T(c) \cdot \phi_r(c)$$

Again, the score $\Delta_s(\bar{e})$ can be found as $\pi^T X w$. Due to the stratified representation of the score, the variance of $\Delta_s(\bar{e})$ breaks down to a weighted sum of the per-stratum variances:

$$var\left[\Delta_s(\bar{e})\right] = \frac{1}{N} \sum_i \pi_i \cdot var_i\left[S\right] = \frac{1}{N} \sum_i \pi_i \cdot w^T Z_i w$$

where N is the number of interactions in $\bar{e}$, and Z_i is the covariance matrix of the interaction scores $\sum_{c \in q} T(c) \cdot \phi(c)$ for the ith stratum:

$$Z_i = \sum_{q \in Q_{\bar{e},i}} \frac{1}{|Q_{\bar{e},i}|} \left(\sum_{c \in q} T(c)\phi(c) - \bar{\phi}_i \right) \left(\sum_{c \in q} T(c)\phi(c) - \bar{\phi}_i \right)^T$$

and $\bar{\phi}_i$ is the mean feature vector for the ith stratum:

$$\bar{\phi}_i = \frac{1}{|Q_{\bar{e},i}|} \sum_{c \in q, q \in Q_{\bar{e},i}} T(c) \cdot \phi(c)$$

Overall, we obtain the following optimization problem:

$$\begin{aligned} &\hat{\pi}, \hat{w} = \arg\max_{\pi, w} \frac{\pi^T X w}{\sqrt{w^T \left(\sum_i \pi_i \cdot Z_i\right) w}} \\ &s.t.\ R1, R2,\ w^T w = 1 \end{aligned} \tag{10}$$

The use of stratification considerably simplifies the form of the optimization problem (10). Indeed, to calculate the variance of $\Delta_s(e)$ in the denominator of Equation (9) we used the right part of the inequality (6). In the non-stratified case, the variance is represented by the left part of (6). The latter case is harder for the optimization due to additional mutual dependencies of the variables (e.g. the variance becomes a third-order polynomial w.r.t. π, while it is linear in the stratified case).

In contrast to the case considered by Yue et al., there is no closed-form solution to the problems (8) and (10) (due to the additional variable π and requirements R1 and R2). Instead, we optimize (8) and (10) numerically[4]. As an initial approximation, we use the uniform policy and the solution of the corresponding problem in [20].

7. DATASETS

In our evaluation study we use two datasets: a dataset of Team Draft-based document search online experiments performed by a commercial search engine, and a dataset of preliminary interleaving experiments performed on the image search service. We discuss them in more detail below.

Document search We build the dataset of the Team Draft-based online experiments as follows. First, we randomly sample a subset of interleaving experiments performed by the search engine in the period from January to November, 2014. These experiments test changes in the search ranking algorithm that were developed as a part of the search engine's evolution. The experiments also differ by country, and geographical region they are deployed on. We select the experiments where the winner (A or B) is determined with a high level of confidence, $p \leq 0.005$ (binomial sign test, deduped click weighting scheme [2]).

Table 1: Datasets statistics.

Domain	# exp.	B > A	mean/median # sessions	mean/median # days
Document	67	30	840K/620K	9.8/8.0
Image	5	0	38K / 34K	4.0/4.0

Image search In contrast to the web document search case, a representative set of online interleaving experiments is not available to us. Instead, we take five "data collection" experiments. In each of these experiments, the evaluated ranker B is obtained by degrading A in a controlled manner. After that, the corresponding "comparison" of A and B is deployed. In these "experiments" the interleaved result pages are obtained by interleaving the ranked lists returned by A and B, as discussed in Section 3, and showing them with the uniform policy (i.e. applying Team Draft). The following modifications of the production ranker to generate the alternative system B were used:

- swapping the results ranked as 1..15 with the results ranked 16..30;
- random permutation of the top-ranked results;
- promoting results with a low resolution;
- setting an important subset of the ranking features to zero;
- randomly ignoring some subsets of the search index.

As a result, we obtained a dataset of experiments, which can be used to adjust the interleaving parameters w and π, as discussed in Section 6. Once the number of organic evaluation experiments have grown, the optimization procedure we propose can be repeated on a more representative dataset.

We provide descriptive statistics of the datasets in Table 1.

8. INSTANTIATION

As discussed above, what changes for different domains is the feature representation of the clicks ($\phi(c)$). Further we describe what features we use in our experimental study. All features we use are independent from the result page demonstrated, so they meet the requirements of Lemma 1.

Document search features For each click in a user interaction, we calculate a set of 24 features, split into four families: Rank-based, Dwell time-based, Order-based, and Linear score-based features. We report these features along with their descriptions in Table 2.

Image search features The click features we use for image search interleaving are similar to the features used for document search. We exclude some rank-based features, as they are not meaningful for the two-dimensional result presentation (e.g. feature #11 assumes that the users tend to examine results in a rank-wise order). The full list of features used for the image search click representation is provided in Table 3.

Stratification In the document search scenario, we stratify the estimate of the experiment outcome according to the teams of the results on the first result page. This gives us $2^{10/2} = 32$ strata. The same strata are used for the policy optimization: the policy specifies the probability of using a specific team distribution to generate the first interleaved

[4] Using the SLSQP routine implemented in scipy [9].

Table 2: Click features for document search.

Feature family	id	Description
Rank-based		Transformations of the click's rank, normalized by the number of clicks
	1-10	position indicators, $f_i = \mathbb{I}\{rank = i\}$
	11	rank
	12	$\sqrt{rank}$
	13	$log(rank)$
	14	$\mathbb{I}\{rank > 4\}$
	15	$\mathbb{I}\{rank > d\}$, where d is the number of identical results in the tops of A and B
Dwell time-based		Indicators of the dwell time (seconds), normalized by the number of clicks
	16	$\mathbb{I}\{dwell \leq 30\}$
	17	$\mathbb{I}\{dwell \in (30, 60]\}$
	18	$\mathbb{I}\{dwell \in (60, 90]\}$
	19	$\mathbb{I}\{dwell \in (90, 120]\}$
	20	$\mathbb{I}\{dwell > 120\}$
Order-based		Indicators of the click's position in the interaction
	21	is the click first
	22	is the click last
Linear score-based		after applying the scoring rule F4, these features represent the (normalized) number of clicks the results from B received
	23	$f_{23} = 1$
	24	$f_{24} = 1/n$, where n is the total number of clicks

result page. The remaining pages are generated using the standard Team Draft procedure, and it can be shown that the interleaving is unbiased in terms of R1 in that case.

In the case of image search, the stratification is less straightforward. Indeed, the stratification according to the teams of the top 30 results on the first result page, will yield $2^{30/2} = 32768$ strata. On one hand, according to Equation (6), using more fine-grained strata results in equal or lower variance. On the other hand, to run the optimization discussed in Section 6, we need to estimate per-stratum means and covariances of the features. This results in a trade-off between an increased sensitivity due to more fine-grained stratification and a higher error of the optimization with unreliable parameters. Thus we performed the search for the optimal number of top results to be used in stratification as a part of the training process, as discussed in Section 9.3.

9. EVALUATION

In our evaluation study, we aim to answer the following research questions: (RQ1) is our framework more sensitive than the baselines on the document and image search data, and (RQ2) if yes, then what aspects of the sensitivity optimization (stratification, credit assignment and policy optimization) contribute to the increased sensitivity?

To answer these questions, we firstly describe the baselines we use in Section 9.1. After that, we introduce the metric we use in Section 9.2. Finally, we describe the evaluation methodology in Section 9.3.

9.1 Baselines

In our study, we compare the sensitivity of our proposed framework to the Team Draft algorithm with the credit assignment functions varied. We consider credit assignment functions of two types: the heuristic click weighting schemes that are applicable for Team Draft and considered in [2], and the learned scoring functions trained according to the approach of Yue et al. [20]. All these baselines are non-stratified.

Linear In the simplest scoring scheme, we calculate the difference in the number of clicks on the results from A and B:

$$S(q; w) = \sum_{c \in q} T(c)$$

Normalized Linear In the *Normalized Linear* scheme, the score of B in a particular interaction is normalised by the number of clicks in this interaction:

$$S(q; w) = \frac{1}{|q|} \sum_{c \in q} T(c)$$

Binary Another approach to aggregate clicks in a single impression is to assign a unit credit to the alternative that received more clicks:

$$S(q; w) = sign\left(\sum_{c \in q} T(c)\right)$$

Deduped Binary In the web document search scenario, it is often assumed that the users examine result lists from top to bottom. In that case, if the top k documents are identical both in A and B, all the interleaved lists have the same top k results, too. Thus, clicks on these top k results add a zero mean additive noise to the difference between the number of clicks A and B receive. A useful trick is to ignore such clicks. We combine this approach with the binary aggregation scheme:

$$S(q; w) = sign\left(\sum_{c \in q} T_d(c)\right)$$

where $T_d(\cdot)$ is a modified team indicator function, equal to zero if the click is performed on one of the top results, identical for A and B, and equal to $T(\cdot)$ otherwise. The deduped binary scheme is one of the most sensitive schemes [2].

Learned-mean, Learned-z In contrast to the above discussed credit assignment functions that are based on intuitive considerations, *Learned-mean* and *Learned-z* are machine-learned credit assignment functions that based on the approach of Yue et al. [20]. These baselines use the same feature representations as our proposed interleaving framework. However, the optimization of the interleaving policy is not performed, and it is fixed to be constant and uniform (as in Team Draft). *Learned-mean* selects the vector of weights w such that the differences between A and B are maximized, and *Learned-z* maximizes the z-score objective. These objectives are close to the objectives we use in Section 6, but they assume a non-stratified experiment outcome and the uniform policy.

It would be interesting to compare our framework to the Optimized Interleaving framework [15]. However, Optimized Interleaving relies on considerably larger sets of interleaved result pages, thus the datasets of Team Draft-based interleaving experiments cannot be re-used to evaluate its performance. An alternative approach is to leverage the natural variation of the search engine's rankings as a source of the result pages, as used in [15]. However, in this case, the evaluation is performed on a query level, and it is restricted to be based on the head queries only. Overall, this might lead to a less representative study.

9.2 Metric

In this work, we use the *z-score* metric that is used to measure the interleaving sensitivity on the historical data [2, 10]. *z-score* indicates the confidence of the evaluated method in the experiment outcome, thus it serves as a proxy to measure

Table 3: Click features for image search.

Feature family	id	Description
Rank-based		position indicators
	1-30	$f_i = \mathbb{I}\{rank = i\}$
	31	$\mathbb{I}\{rank > d\}$, where d is the number of identical results in the tops of A and B
Dwell time-based		Indicators of the dwell time (seconds), normalized by the number of clicks
	32	$\mathbb{I}\{dwell \leq 30\}$
	33	$\mathbb{I}\{dwell \in (30, 60]\}$
	34	$\mathbb{I}\{dwell \in (60, 90]\}$
	35	$\mathbb{I}\{dwell \in (90, 120]\}$
	36	$\mathbb{I}\{dwell > 120\}$
Order-based		Indicators of the click's position in the interaction
	37	is the click first
	38	is the click last
Linear score-based		after applying the scoring rule F4, these features represent the (normalized) number of clicks the results from B received
	39	$f_{48} = 1$
	40	$f_{49} = 1/n$, where n is the total number of clicks

the sensitivity of the method: a higher confidence indicates a higher sensitivity.

Assuming that $\Delta_s(e)$ is normally distributed[5] and using the notation introduced above, we define the *z-score* statistic on the data of the experiment e as follows:

$$Z = \frac{\Delta_s(e)}{\sqrt{var[\Delta_s(e)]}} = \frac{\Delta_s(e)}{\sqrt{\sum_i \pi_i \cdot var_i[S]}}\sqrt{N} \qquad (11)$$

To calculate the z-score statistic for an interleaving method with a non-uniform policy on data obtained from an experiment with the uniform policy, we use the per-stratum sample estimates of the expectation $\mathbb{E}_i[S]$ and the variance $var_i[S]$ (Equation (6)), calculated on the experimental data, and the policy specified by the interleaving method.

The value of (11) indicates how far the score $\Delta_s(e)$ deviates from zero in the standard normal distribution. Thus it indicates the confidence level of the experiment outcome and can be mapped into p-value (under the null hypothesis the true value of $\Delta_s(e)$ is 0). For instance, Z of 1.96 (2.58) corresponds to the two-sided p-value of 0.05 (0.01).

In the case of the non-stratified estimate $\Delta(e)$, z-score is calculated similarly:

$$Z = \frac{\Delta(e)}{\sqrt{var[\Delta(e)]}} = \frac{\Delta(e)}{\sqrt{var[S]}}\sqrt{N} \qquad (12)$$

For each interleaving experiment, we calculated the relative z-score by dividing the outcome's z-score by the z-score of the Team Draft method with the linear click weighting scheme. The relative z-score z_e has an intuitive interpretation [2]: the corresponding interleaving method needs z_e^2 less interactions in the same experiment e than the Team Draft algorithm with the linear weighting scheme to achieve the same level of confidence.

9.3 Procedure

In our evaluation on the document search dataset, we use 10-fold cross-validation: in each split, 90% of the interleaving experiments are used for optimization, and 10% are used to evaluate the resulting sensitivity. The same splits are used for all the approaches that run optimization (our proposed framework with two types of dissimilarity, and *Learned-mean* and *Learned-z* baselines). In each split, we measure the relative z-scores of an interleaving method on the experiments in the test set. For each interleaving method, we report the overall mean and the median relative z-scores collected across all folds. We use the paired t-test on the absolute values of the non-normalized z-scores when testing the statistical significance of the performance differences.

In the case of image search, due to the smaller dataset, we replace the 10-fold cross-valuation with the leave-one-out procedure: one experiment is used for evaluation, while the others are used for training. Further, within a training step, we additionally run a nested 2-fold cross-validation procedure on the training set to find the optimal number of the result teams to be considered in stratification: for k in $3, ..., 15$ we evaluate the performance of our proposed method when teams of the top $2k$ are used for the stratification. The search is stopped when the performance degrades. In most folds the optimal k is found to be equal to 3 (i.e., the top 6 results are used for the stratification).

10. RESULTS AND DISCUSSION

In this section, we use the following notation. Linear, Normalized Linear, Binary, and Deduped Binary weighting schemes correspond to *Linear, NLinear, Binary*, and *Deduped*, respectively. L_m and L_z indicate the Learned-mean and Learned-z baselines. The instantiations of our proposed framework are referred to as F_m and F_z, when the optimization is performed to maximize the mean difference (8) and the z-score (10) objectives, respectively.

As we are interested in evaluating the effects of the stratification and the effects of the joint optimization individually, we additionally measure the performance of the baselines when the stratified outcome $\Delta_s(e)$ is calculated. The stratified modifications of the interleaving methods L_m and L_z are denoted as L_m^s and L_z^m. L_m^s and L_z^s use the stratified objectives we proposed in Section 6, and correspond to our framework with the interleaving policy fixed to be uniform.

In our experiments on both document and image search datasets, all of the studied interleaving methods correctly determined the preference for A or B.

10.1 Document Search

In Table 4 we report the results of the evaluation procedure discussed in Section 9.3 applied for the web document search data. In the left part of Table 4 (*Non-stratified* column), we report the mean and median relative z-scores for the baselines with no stratification applied. In the right part (*Stratified* column), we report the performance of our proposed framework as well as for the baselines with the stratification applied.

On analysing the results of the non-stratified baselines, reported in the left part of Table 4, we notice that their relative performance is generally in line with the results reported in [2]. Indeed, the deduped binary scheme with its median relative z-score of 1.59 considerably outperforms other considered heuristic schemes: Linear (1.0), Normalized Linear (0.93), and Binary (0.98); similarly, L_z outperforms L_m.

On comparing the relative z-scores of Linear, NLinear, Binary, and Deduped with and without the stratification applied (left vs right parts of Table 4), we observe that in some cases the stratification greatly increases the interleaving sensitivity. For instance, the mean and the median relative z-scores of Binary grow from 1.10 and 0.98 to 1.22 and

[5] This assumption holds when $\pi_i \cdot N$ is large enough for all i with $\pi_i > 0$, as $\Delta_s(e)$ is a sum of approximately normally distributed per-stratum sample means, thus it is normally distributed.

Table 4: Relative confidence levels of the interleaving outcomes, document search. The scores of the interleaving method with the highest sensitivity ($p < 0.01$) are denoted by $^\diamond$.

	Non-stratified						Stratified							
	Linear	NLinear	Binary	Deduped	L_m	L_z	Linear	NLinear	Binary	Deduped	L^s_m	L^s_z	F_m	F_z
Mean	1.00	1.03	1.10	1.88	1.34	2.14	1.06	1.16	1.22	1.88	1.39	2.28	1.38	**2.45**$^\diamond$
Median	1.00	0.93	0.98	1.59	1.20	1.80	1.04	1.03	1.10	1.60	1.24	1.96	1.23	**2.05**$^\diamond$

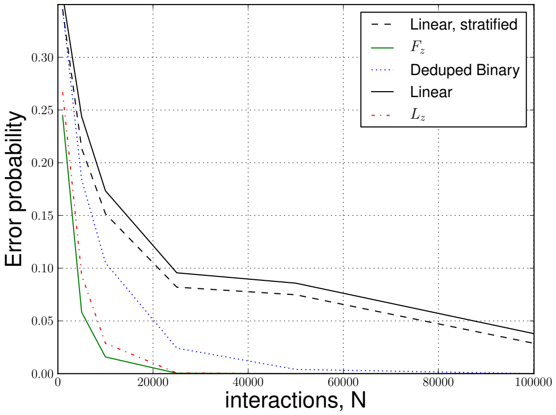

Figure 1: The probability that an interleaving method disagrees with the true preference, depending on the size of the sample.

1.10, respectively. Noticeable improvements are obtained for all of the considered baseline schemes, except for Deduped, where the improvement is small. Interestingly, a considerable improvement is also observed for L_z: its stratified modification L^s_z exhibits a median relative z-score of 1.96, while L_z has a median z-score of 1.80. This level of improvement is roughly comparable to the difference between the best heuristic baseline (Deduped, median relative z-score 1.59) and the best machine-learned baseline (L_z, median relative z-score 1.80) in the non-stratified case.

In all cases, the credit assignment function that optimizes the mean difference between A and B performs worse the credit assignment functions learned to maximize the z-score. For instance, L^s_z demonstrates considerably higher median relative confidence than L^s_m (1.96 vs 1.24).

By additionally performing the interleaving policy optimization, F_z achieves a considerable sensitivity gain in comparison with the stratified L^s_z (F_z, 2.05 vs L_z, 1.96). This gain is roughly similar to the difference between performance obtained by performing stratification (L_z, 1.80 vs L^s_z, 1.96). F_z also achieves the highest overall sensitivity, with the median relative z-score of 2.05 and the mean z-score of 2.45. This implies that an interleaving experiment that uses the interleaving method F_z requires $2.05^2 = 4.20$ times less user interactions (in median) than the non-stratified Team Draft with the linear scoring to achieve the same level of confidence. In comparison with the best performing baseline, L_z, it requires $(\frac{2.05}{1.80})^2 = 1.30$ times less data to achieve the same level of confidence (in median).

Visualization We illustrate the relative performance of the studied interleaving methods on the document search dataset using the following procedure. We randomly select one experiment to be used as a test experiment, and use the remaining experiments to optimize the interleaving parameters. Further, we estimate the probability that an interleaving method disagrees with the ground truth preference in the test experiment by obtaining 10,000 samples of N user interactions. We varied N in $(10^3, ..., 10^5)$. For the baseline methods, N interactions are obtained by sampling from the experiment's interactions with replacement (bootstrap sampling). For F_z, the sample is obtained by firstly allocating N interactions to the strata according to multinomial distribution specified by the policy π, and further sampling from the individual strata (with replacement). The outcome is calculated using the stratified estimate Δ_e. This sampling process simulates the case of policy π to be applied in a real-life scenario. The stratified modification of *Linear* differs from *Linear* only by using the stratified estimate of the outcome, Δ_s instead of the sample mean Δ. A higher error probability indicates lower sensitivity and it is related to the outcome's p-value under the bootstrap test.

In Figure 1, we report the obtained error probabilities. From Figure 1 we observe that the optimization-based methods (F_z and L_z) dramatically increase the interleaving sensitivity and outperform *Linear*, stratified *Linear*, and *Deduped* by a considerable margin. For instance, the probability error of 0.05 is achieved by F_z with less than 10,000 interactions, but *Linear* requires about 90,000 interactions to achieve the same level of error. Among the methods that use optimization, F_z consistently demonstrates lower probability of error than L_z. For instance, when $5,000$ interactions is used, F_z has the probability of error below 0.06, while L_z makes an error in more than 0.09 of the samples. Further, we observe that the performance of *Linear* is noticeably improved by adding stratification. Overall, these observations are in line with results reported in Table 4. However, this illustration is also important as it does not rely on the z-score statistic.

10.2 Image Search

In Table 5 we report the results of the evaluation for the case of image search. Generally, we observe the results similar to the document search case. The machine-learned interleaving methods that optimize the z-score objective (L_z, L^s_m, L^s_z, and F_z) outperform both the methods with the heuristic credit assignment (*Linear*, *NLinear*, *Binary*, and *Deduped*) and the methods that optimize the mean difference.

In contrast to the document search experiments, the sensitivity gains due to stratification are less noticeable on the image search data. A possible explanation is that the differences of the means of the strata are smaller than in the case of the document search. Indeed, if the users tend to examine most of the results (which is easier for images than for document snippets) and click more, then the teams of the first results is not such a strong indicator of the total credit in an interaction, as in the case of document search. Interestingly, *Deduped* is sensitive in image search, too.

The overall highest performance ($p < 0.05$) is achieved by our proposed framework with the z-score-based optimization objective (F_z, mean relative z-score 1.21, median 1.18). This value of the metric implies that our proposed framework requires $1.18^2 = 1.39$ less data (median) than the *Linear* baseline to achieve the same level of confidence. In comparison to the best-performing baseline L_z, the corresponding de-

Table 5: Relative confidence levels of the interleaving outcomes, image search. The scores of the interleaving method with the highest sensitivity ($p < 0.05$) are denoted by $^\diamond$.

	Non-stratified						Stratified							
	Linear	NLinear	Binary	Deduped	L_m	L_z	Linear	NLinear	Binary	Deduped	L^s_m	L^s_z	F_m	F_z
Mean	1.00	1.03	1.05	1.17	1.11	1.17	1.00	1.03	1.05	1.17	1.11	1.18	1.14	**1.21**$^\diamond$
Median	1.00	0.98	1.02	1.11	1.08	1.16	1.00	0.98	1.03	1.11	1.08	1.16	1.14	**1.18**$^\diamond$

crease is $(\frac{1.18}{1.16})^2 = 1.03$ in median. However, the difference of the means is higher (L_z, 1.18 vs F_z, 1.21). A possible explanation for the smaller improvements is that the degradations used in our image search dataset are relatively strong and easy to detect, thus it is harder achieve a high level of improvement over the baselines.

Summary Our evaluation study allows us to answer the research questions we stated in Section 9. Our proposed framework achieves the highest sensitivity on both the document search and the image search datasets (RQ1). On the document search data, each of the proposed sensitivity optimization aspects contributes to the increased performance the credit optimization. Indeed, the non-stratified optimized interleaving method L_z (median, 1.80) outperforms best of the non-learned baselines (*Deduped Binary*, 1.59). In turn, the stratified method L^s_z has even higher performance (1.96). Further, our proposed F_z additionally performs the policy optimization and achieves the highest median relative confidence level (2.05). In contrast, on the image search data the contribution of the stratification is small. However, gains in the sensitivity are still obtained by the credit assignment learning (L_z, 1.16 vs *Deduped*, 1.11) and the policy optimization (F_z, 1.18 vs L^s_z, 1.16). These observations answer RQ2: the credit assignment and the policy optimization increase the interleaving sensitivity on both datasets; our proposed stratification has higher impact on the interleaving sensitivity in the document search than in the image search domain.

11. CONCLUSION

In this work we address an important problem of improving of the interleaving sensitivity. We proposed an interleaving framework that generalizes the existing research in two aspects. First, it achieves an increased sensitivity by performing a joint optimization of the credit assignment function and the interleaving policy. Second, it is formulated to be general with respect to the way the results are presented, thus it can be applied in the domains with the grid-based representation, such as image search. Further, to simplify the optimization procedure, we proposed to use a stratified estimate of the experiment outcome. This stratification is useful on its own, as in some cases it reduces the variance of the experiment outcome and thus increases the sensitivity. Finally, we proposed a generalized unbiasedness requirement that the feature-based credit assignment and the interleaving policy have to meet for the interleaving to be unbiased.

In our evaluation study, we used two datasets of the Team Draft-based experiments obtained from a commercial search engine. The first dataset contains 67 interleaving experiments performed in the document search domain, and the second dataset contains 5 “data collection” experiments deployed for image search. In our study, we demonstrated that our proposed framework achieves the highest sensitivity on both datasets. Specifically, we observe that our framework requires up to 1.30 times (median) less data than the top-performing baseline on the document search dataset, and up to 1.03 times (median) less data on the image search dataset.

An interesting direction of future work is to apply our framework to other domains such as video search, and to incorporate a non-linear credit assignment function in our proposed framework.

Acknowledgements We thank Aleksandr Vorobev for his feedback on a draft of this paper.

12. REFERENCES

[1] S. Asmussen and P. W. Glynn. *Stochastic simulation: Algorithms and analysis*, volume 57. Springer Science & Business Media, 2007.

[2] O. Chapelle, T. Joachims, F. Radlinski, and Y. Yue. Large-scale validation and analysis of interleaved search evaluation. *ACM TOIS*, 30(1):6, 2012.

[3] A. Chuklin, A. Schuth, K. Hofmann, P. Serdyukov, and M. de Rijke. Evaluating aggregated search using interleaving. In *CIKM 2013*.

[4] A. Deng, Y. Xu, R. Kohavi, and T. Walker. Improving the sensitivity of online controlled experiments by utilizing pre-experiment data. In *WSDM 2013*.

[5] G. Dupret and M. Lalmas. Absence time and user engagement: evaluating ranking functions. In *WSDM 2013*.

[6] K. Hofmann, S. Whiteson, and M. de Rijke. A probabilistic method for inferring preferences from clicks. In *CIKM 2011*.

[7] T. Joachims. Optimizing search engines using clickthrough data. In *KDD 2002*.

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

[9] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.

[10] E. Kharitonov, C. Macdonald, P. Serdyukov, and I. Ounis. Using historical click data to increase interleaving sensitivity. In *CIKM 2013*.

[11] R. Kohavi, T. Crook, R. Longbotham, B. Frasca, R. Henne, J. L. Ferres, and T. Melamed. Online experimentation at microsoft. *Data Mining Case Studies*, page 11, 2009.

[12] R. Kohavi, A. Deng, B. Frasca, T. Walker, Y. Xu, and N. Pohlmann. Online controlled experiments at large scale. In *KDD 2013*.

[13] R. Kohavi, R. Longbotham, D. Sommerfield, and R. M. Henne. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*, 18(1), 2009.

[14] F. Radlinski and N. Craswell. Comparing the sensitivity of information retrieval metrics. In *SIGIR 2010*.

[15] F. Radlinski and N. Craswell. Optimized interleaving for online retrieval evaluation. In *WSDM 2013*.

[16] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? In *CIKM 2008*.

[17] C. Robert and G. Casella. *Introducing Monte Carlo Methods with R.* Springer Science & Business Media, 2009.

[18] A. Schuth, F. Sietsma, S. Whiteson, D. Lefortier, and M. de Rijke. Multileaved comparisons for fast online evaluation. In *CIKM 2014*.

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

[20] Y. Yue, Y. Gao, O. Chapelle, Y. Zhang, and T. Joachims. Learning more powerful test statistics for click-based retrieval evaluation. In *SIGIR 2010*.