A Probabilistic Method for Inferring Preferences from Clicks

Katja Hofmann, Shimon Whiteson and Maarten de Rijke
ISLA, University of Amsterdam
{K.Hofmann, S.A.Whiteson, deRijke}@uva.nl

ABSTRACT

Evaluating rankers using implicit feedback, such as clicks on documents in a result list, is an increasingly popular alternative to traditional evaluation methods based on explicit relevance judgments. Previous work has shown that so-called *interleaved comparison methods* can utilize click data to detect small differences between rankers and can be applied to learn ranking functions online.

In this paper, we analyze three existing interleaved comparison methods and find that they are all either biased or insensitive to some differences between rankers. To address these problems, we present a new method based on a probabilistic interleaving process. We derive an unbiased estimator of comparison outcomes and show how marginalizing over possible comparison outcomes given the observed click data can make this estimator even more effective.

We validate our approach using a recently developed simulation framework based on a learning to rank dataset and a model of click behavior. Our experiments confirm the results of our analysis and show that our method is both more accurate and more robust to noise than existing methods.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.3 Information Search and Retrieval

General Terms

Algorithms, Experimentation

Keywords

Evaluation, Implicit feedback, Interleaved comparison

1. INTRODUCTION

In information retrieval (IR), most techniques for evaluating rankers require *explicit feedback*, in which assessors manually judge the relevance of documents for given queries. Unfortunately, obtaining such explicit judgments is expensive and error-prone. Since annotators usually judge documents for queries they did not formulate, the judgments may be biased towards their interpretation of the underlying information needs [2]. Consequently, evaluating rankers using *implicit feedback* such as click data is a promising alternative. Since click data is a by-product of the normal interactions between the user and the retrieval system, it can be collected unobtrusively and at minimal cost. In addition, since it is based on the behavior of real users, it more accurately reflects how well their actual information needs are met [23].

Previous work demonstrated that two rankers can be successfully compared using click data. In the *balanced interleave* method [15], an *interleaved list* is generated for each query based on the two rankers. The user's clicks on the interleaved list are attributed to each ranker based on how they ranked the clicked documents, and the ranker that obtains more clicks is deemed superior. These comparisons are repeated for a large number of queries, to obtain a reliable estimate of the relative performance of the two rankers.

Unfortunately, the balanced interleave approach suffers from a bias that can make it prefer the wrong ranker [23]. The alternative *team draft* method avoids this form of bias by assigning each document to the ranker that contributed it to the interleaved list and only crediting clicks to the ranker to which the clicked document is assigned. In contrast to the balanced interleave and team draft methods, the *document constraint* method takes relations between documents into account [10].

In this paper, we analyze existing interleaved comparison methods and find that the document constraint method suffers from a bias similar to that of the balanced interleave method. We also find that the team draft method, though unbiased, is insensitive to some differences between rankers. We illustrate these problems using examples, and show how they can affect comparison outcomes.

Furthermore, we propose a new approach for comparing rankers that utilizes a probabilistic model of the interleave process and is based on the team draft method. The main idea is to replace the original rankers with softmax functions that define probability distributions over documents. Contributing a document to an interleaved list is formulated as sampling without replacement from a probability distribution. To infer comparison outcomes, we derive an unbiased estimator. Because this estimator is based on our probabilistic model, it smooths over click assignments, making the method both unbiased and sensitive to small differences between rankers. Finally, we show how comparisons can make more effective use of observed data by marginalizing over all possible comparison outcomes for each observed sample.

We present experiments using a large set of explicitly labeled data and a recently developed user click model that simulates the resulting implicit feedback. These experiments confirm the results of our analysis, and show that our method can identify the correct ranker more reliably than existing methods. Furthermore, our method improves over previous methods in terms of efficiency and robustness to noise.

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

In the next section we give an overview of related work (§2). We then formulate requirements for interleaved comparison methods and analyze existing methods (§3). Our probabilistic method is described in §4 and experiments are detailed in §5. We present and discuss experimental outcomes in §6 and conclude in §7.

2. RELATED WORK

Implicit feedback has long been used in a variety of settings, e.g., to infer users' interest in specific web pages. Many forms have been considered, such as the time spent on a page, scrolling, and clicks [5, 17]. In IR, implicit feedback has been used for relevance feedback, to refine search results throughout a search session [27].

Unfortunately, implicit feedback tends to be noisy. For example, a long-term study of "time spent reading" found no clear relation between this form of implicit feedback and a document's perceived usefulness, and that the time spent on the page varies considerably across users and tasks [18].

Clicks, the most easily collected form of feedback, are difficult to use for evaluation because they do not correspond well to absolute relevance. Jung et al. [16] found that click data does contain useful information, but that variance is high. They propose aggregating clicks over search sessions and show that focusing on clicks towards the end of sessions can improve relevance predictions. Similarly, Scholer et al. [24] found that click behavior varies substantially across users and topics, and that click data is too noisy to serve as a measure of absolute relevance. Fox et al. [7] found that combining several implicit indicators can improve accuracy, though it remains well below that of explicit feedback.

Nonetheless, in some applications, click data has proven reliable. In searches of expert users who are familiar with the search system and document collection, clicks can be as reliable as purchase decisions [11]. Methods for optimizing the click-through rates in ad placement [19] and web search [22] have also learned effectively from click data.

Methods that use implicit feedback to infer the relevance of specific document-query pairs have also proven effective. Shen et al. [25] show that integrating click-through information for query-document pairs into a content-based retrieval system can improve retrieval performance substantially. Agichtein et al. [1] demonstrate dramatic performance improvements by re-ranking search results based on a combination of implicit feedback sources, including click-based and link-based features. Chapelle and Zhang [4] learn a Bayesian model that accurately predicts the relevance of documents to specific queries from click data.

These applications learn the value of specific documents for specific queries, for which implicit feedback appears to be accurate. However, since implicit feedback varies so much across queries, it is difficult to use it to learn models that generalize across queries. To address this problem, several methods have been developed that use implicit feedback, not to infer absolute judgments, but to compare two rankers by observing clicks on an interleaved result list. In the remainder of this section, we overview these methods.

The *balanced interleave* method [15] takes as input two result lists l_1 and l_2 for a given query q, and produces an outcome $o \in \{-1, 0, 1\}$ that indicates whether the quality of l_1 is judged to be lower, equal to, or higher than that of l_2, respectively. The method first generates an interleaved list l (see Figure 1, part 1). One of $\{l_1, l_2\}$ is selected at random as the starting list and its first document is placed at the top of the interleaved list. Then, the non-starting list contributes its highest-ranked document if it is not already part of the list. These steps repeat until the interleaved list has the desired length. The clicks c that are observed when the interleaved list is presented to the user are attributed to each list as

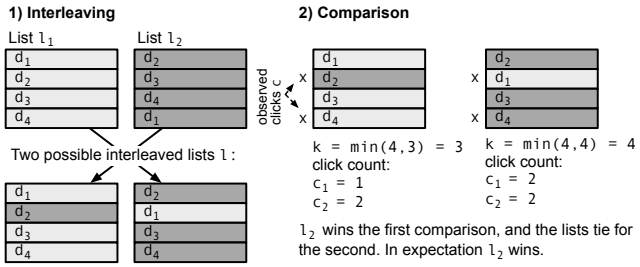

Figure 1: Interleaved comparisons using *balanced interleave*.

follows (Figure 1, part 2). For each list, the lowest-ranked document that received a click is determined, and the minimum of their ranks is denoted k. Then, the clicked documents ranked at or above k are counted for each list. The list with more clicks in its top k is deemed superior (they tie if they obtain the same number of clicks).

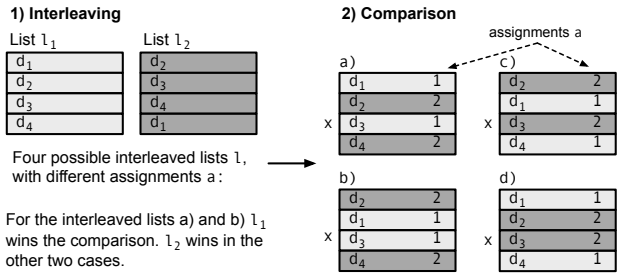

Figure 2: Interleaved comparisons using *team draft*.

The alternative *team draft* method [23] creates an interleaved list following the model of "team captains" selecting their team from a set of players (see Figure 2). For each pair of documents to be placed on the interleaved list, a coin flip determines which list gets to select a document first. It then contributes its highest-ranked document that is not yet part of the interleaved list. The method also records which list contributed which document in an *assignment* a. To compare the lists, only clicks on documents that were contributed by each list are counted towards that list. This method of assignments ensures that each list has an equal chance of obtaining clicks. Recent work demonstrates that the team draft method can reliably identify the better of two rankers in practice [21].

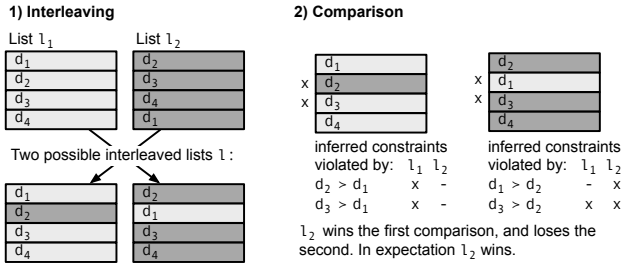

Figure 3: Interleaved comparisons using *document constraints*.

Neither the balanced interleave nor the team draft method takes relations between documents explicitly into account. To address this, He et al. [10] propose an approach that we call the *document-constraint* method (see Figure 3). Result lists are interleaved and clicks observed as for the balanced interleave method. Then, following [14], the method infers constraints on pairs of individual documents, based on their clicks and ranks. For each pair of a clicked document and a higher-ranked non-clicked document, a

constraint is inferred that requires the former to be ranked higher than the latter.[1] For example, in the first interleaved list shown in Figure 3, d_2 is ranked lower than d_1. If d_2 is clicked while d_1 is not, the constraint $d_2 \succ d_1$ is inferred. The method then compares these constraints to the original result lists and counts how many constraints are violated by each. The list that violates fewer constraints is deemed superior. Though more computationally expensive, this method proved more reliable than either balanced interleave or team draft on a synthetic data set [10].

3. ANALYSIS

To analyze interleaved comparison methods, we need a set of criteria against which such methods can be evaluated. We propose two criteria. First, an interleaved comparison method should not suffer from *bias*, i.e., it should not prefer either ranker when clicks are random. Second, it should possess *sensitivity*, i.e., it should be able to detect differences in the quality of document rankings. In this section, we detail both criteria and analyze to what degree they are met by existing methods.

3.1 Bias

Our first criterion for interleaved comparison methods is that they not suffer from *bias*. We define an interleaved comparison method to be biased if, under a random distribution of clicks, it prefers one ranker over the other in expectation. Bias is problematic because it means that noise in click feedback can affect the expected outcome of comparisons, making the comparison method unreliable.

For the balanced interleave method, Radlinski [23] demonstrated bias in cases where l_1 and l_2 are very similar. This problem is illustrated in Figure 1. The interleave process can result in interleaved lists that are identical, or very similar, to one of the original lists. In the example, the first interleaved list is equal to l_1, and the second is similar to it, with only the first two documents switched. Consequently, clicks towards the top of the interleaved list are more likely to be attributed to l_1, while clicks on lower-ranked documents tend to benefit l_2. This is problematic for two reasons. First, the biases for either list do not necessarily cancel each other out, so that even randomly distributed clicks can yield a preference for one of the lists. For example, in Figure 1, l_2 wins in expectation, i.e., the outcome converges to l_2 in the limit; this can easily be verified using truth tables. Second, documents displayed at the top of a result list are typically more likely to be clicked, due to position bias [6]. In the example, strong position bias would favor l_1.

For the document constraint method, we identify a similar bias. Since interleaving is done as in the balanced interleave method, the interleaved lists can again be very similar to one of the original lists. Here, the list that is more similar to a presented interleaved list has a disadvantage because it is more likely to attract clicks. These clicks translate into constraints that are more likely to be violated by the more similar list. For example, in the first interleaved list constructed in Figure 3, clicks on the second and third rank of the interleaved list result in two violated constraints for the similar l_1 and no violated constraints for l_2. For randomly distributed clicks, l_2 wins in expectation.

The team draft method addresses the issue of bias by ensuring that each click can only be attributed to one list (the one assigned to that document). Because the interleaving process also ensures that in expectation each list contributes the same number of documents to each rank, the method is unbiased, both under random clicks and in the face of position bias.

[1] In the original description in [10], additional constraints are inferred between the lowest-clicked document and its successor. To simplify analysis, we consider only the first type of constraints. The additional requirements do not qualitatively affect our analytical or experimental results.

Table 1: Notation used throughout this paper. Uppercase letters indicate random variables and lowercase letters indicate the values they take on.

Symbol	*Description*
q	query
d	document
l	interleaved list
r	rank of a document in a document list
a	assignment, a vector of length $\|l\|$ where each element $a[r] \in \{1, 2\}$ indicates whether the document at rank r of an interleaved list l, $l[r]$ was contributed by l_1 or l_2 (or by softmax functions s_1 or s_2, respectively)
c	a vector of clicks observed on an interleaved list l
s_x	softmax function, cf., §4.1, Equation 1
$o(c, a)$	outcome of a comparison computed from clicks c and assignments a

3.2 Sensitivity

Our second criterion for assessing interleaved comparison methods is *sensitivity*. We define sensitivity as the ability of an interleaved comparison method to detect differences in the quality of rankings. As in standard evaluation measures like Normalized Discounted Cumulative Gain (NDCG) [13], we assume that the quality of a ranking depends on how highly it ranks relevant documents.

Clearly, all interleaved comparison methods will fail to be sensitive if user clicks are not correlated with relevance. However, in some cases, the team draft method can suffer from insensitivity even when such correlations exist. Figure 2 illustrates this phenomenon. Assume that d_3 is the only relevant document, and is therefore more likely to be clicked than other documents. Because l_2 ranks this document higher than l_1 it should win the comparison. To compare l_1 and l_2 using the team draft method, four possible interleaved lists can be generated, all of which place document d_3 at the same rank. In two interleaved lists, d_3 is contributed by l_1, and in two it is contributed by l_2. Thus, in expectation, both lists obtain the same number of clicks for this document, yielding a tie. Therefore, the method fails to detect the preference for l_2.

Hence, since the team draft method is insensitive to some ranking differences, it can fail to detect a difference between rankers or infer a preference for the wrong ranker. In contrast, the balanced interleave and document constraint methods can detect these differences in ranking, but because of the bias discussed above, small differences between rankers can result in erratic behavior.

4. METHOD

As discussed above, existing interleaved comparison methods are either biased or insensitive to certain differences between rankings. In this section we develop a probabilistic method that fulfills both criteria. We do so by devising a probabilistic model in §4.1, which is based on the team draft method. Instead of interleaving documents deterministically, we model the interleaving process as random sampling from *softmax* functions that define probability distributions over documents. This model allows us to derive an estimator in §4.2 that is both unbiased and sensitive to even small ranking changes. Finally, in §4.3, we derive a second estimator that marginalizes over all possible assignments to make estimates more reliable. Table 1 shows notation used throughout this section.

4.1 Probabilistic interleaving

We propose a probabilistic form of interleaving in which the interleaved document list l is constructed, not from fixed lists l_1 and l_2, but from *softmax* functions s_1 and s_2 that depend on the query q. The use of softmax functions is key to our approach, as it ensures that every document has a non-zero probability of being selected by each ranker. As a result, the distribution of credit accumulated for clicks is smoothed, based on the relative rank of the document in the original result lists. If both rankers place a given document at the same rank, then the corresponding softmax functions have the same probability of selecting it and thus they accumulate the same number of clicks in expectation.

More importantly, rankers that put a given document at similar ranks receive similar credit in expectation. Furthermore, the difference in expected value reflects the magnitude of the difference between the two rankings. In this way, the method becomes sensitive to even small differences between rankings and can accurately estimate the magnitude of such differences.

Starting with the ranked lists l_1 and l_2 that two rankers generated for a given query q, we construct corresponding softmax functions s_1 and s_2. Each s_x is a probability distribution over documents in which documents with a higher rank according to l_x have a higher probability. Many softmax functions are possible [20, 26]. We use a function in which the probability of selecting a document is inversely proportional to a power of the rank $r_x(d)$ of a document d in list l_x:

$$P_{s_x}(d) = \frac{\frac{1}{r_x(d)^\tau}}{\sum_{d' \in D} \frac{1}{r_x(d')^\tau}}, \tag{1}$$

where D is the set of all ranked documents, including d. The denominator applies a normalization to make probabilities sum to 1. Because it has a steep decay at top ranks, this softmax function is suitable for an IR setting in which correctly ranking the top documents is the most important. It also has a slow decay at lower ranks, preventing underflow in calculations. The parameter τ controls how quickly selection probabilities decay as rank decreases, similar to the Boltzmann temperature in the normalized exponential function [26]. In our experiments, $\tau = 3$, though preliminary experiments with higher τ yielded qualitatively similar results.

After constructing s_1 and s_2, l is generated similarly to the team draft method. However, instead of randomizing the ranker to contribute the next document per pair, one of the softmax functions is randomly selected at each rank. Doing so is mathematically convenient, as the only component that changes at each rank is the distribution over documents. The system records which softmax function was selected to contribute the next document in assignment a. Then, a document is randomly sampled without replacement from the selected softmax function and added to the interleaved list. The document is also removed from the non-sampled softmax function and both are renormalized. This process repeats until l has the desired length.

This approach has several properties that ensure no bias is introduced. First, in expectation, each ranker contributes the same number of documents to the interleaved list. Second, the softmax functions constructed for each ranker have the same shape, i.e., the probability allocated to each document depends only on its rank. Third, the use of assignments guarantees that each click is attributed to only one list, as in the team draft method.

Figure 4 depicts a probabilistic model of this process. It has four random variables: the user's query q, the interleaved list l generated in response to q, the assignments a that specify which softmax function contributed which document to l, and the user's clicks c. Samples of all four random variables can be observed. In addition,

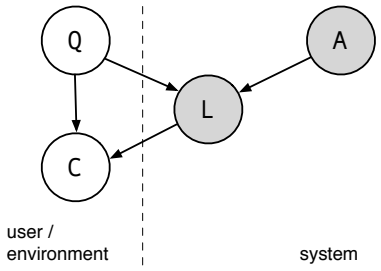

Figure 4: Probabilistic model for comparing rankers. Conditional probability tables are known only for variables in gray.

the distributions L and A are completely specified given a query q. In the next section we show how observed samples (q, l, a, c) can be used to estimate comparison outcomes.

4.2 Unbiased comparison

If we define a deterministic outcome function $o(c, a) \in \{-1, 0, 1\}$ that compares the number of clicks obtained by each ranker, as in the team draft method, then the relative quality of the two rankers is given by the expected outcome, defined as follows:

$$E[o(C, A)] = \sum_{c \in C} \sum_{a \in A} o(c, a) P(c, a). \tag{2}$$

While this cannot be computed directly, it can be estimated from a given set of n samples of the form (c, a):

$$E[o(C, A)] \approx \frac{1}{n} \sum_{i=1}^{n} o(c_i, a_i), \tag{3}$$

where c_i and a_i are vectors containing the observed clicks and assignments for the i-th sample.

Eq. 3 is an unbiased estimator of the expected outcome. If this estimator was used with deterministic ranking functions, it would reduce to the team draft method. However, because of the smooth credit assignment enabled by the softmax functions, it is more sensitive to differences in ranking quality than the team draft method, while still avoiding bias.

4.3 Marginalized comparison

In this section, we derive a second unbiased estimator that achieves better reliability by marginalizing over all possible assignments for an observed interleaved list, instead of using noisy samples from the true distribution.

In Eq. 3, the expected outcome is estimated directly from the observed clicks and assignments. However, in the probabilistic model shown in Figure 4, the distributions for A and L are known given an observed query q. As a result, we need not consider only the observed assignments. Instead, we can consider all possible assignments that could have co-occurred with each observed interleaved list. i.e., we can marginalize over all possible values of A for each observed interleaved list l_i and query q_i. Marginalizing over A leads to the following alternative estimator:

$$E[o(C, A)] \approx \frac{1}{n} \sum_{i=1}^{n} \sum_{a \in A} o(c_i, a) P(a | l_i, q_i), \tag{4}$$

To estimate $P(A|L, Q)$ we can follow Bayes' rule:

$$P(A|L, Q) = \frac{P(L|A, Q) P(A|Q)}{\sum_{a \in A} P(L|a, Q)}. \tag{5}$$

Since A and Q are independent, $P(A|Q) = P(A)$. $P(L|A, Q)$ is the product of the probabilities of selecting the documents at each

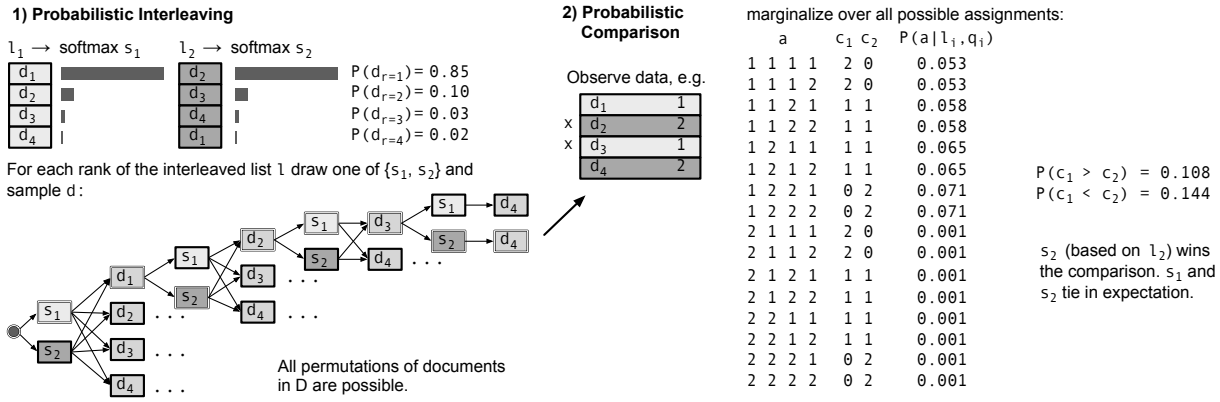

Figure 5: Example of an interleaved comparison using our *probabilistic marginalization* method with softmax functions.

rank:

$$P(L|A,Q) = \prod_{r=1}^{|L|} P(L[r] \mid A[r], L[1,r-1], Q), \qquad (6)$$

where $A[r] \in \{1,2\}$ specifies which softmax function contributed $L[r]$, the document at rank r, and $L[1,r-1]$ is the list of all documents between ranks 1 and $r-1$.

Figure 5 illustrates the resulting probabilistic method for comparing rankers based on implicit feedback. For an incoming query, the system generates an interleaved document list, as described in §4.1. After observing clicks, the system computes the probability of each possible outcome. All possible assignments are generated, and the probability of each is computed using Eq. 5. The example shows that, while the softmax functions tie in expectation (because our method is unbiased), s_2 has a higher probability of winning, because it has a higher probability of generating the clicked documents than s_1 (because the method is sensitive to such differences in rankings).

Generating all possible assignments and computing their probabilities for a given list is computationally expensive. However, assignments and probabilities need to be calculated only up to the lowest observed click. In the worst case, the lowest click is at the lowest rank of the interleaved list, which is a small constant, e.g., 10.[2] In practice, we perform computations using logs and sums instead of probabilities and products to avoid buffer underflow.

5. EXPERIMENTS

We present three sets of experiments, designed to answer the question: *To what extent do the analytical differences between the comparison methods translate into performance differences?*

In our first set of experiments, we address this question on a large scale. We compare a large number of ranker pairs to examine how the analytical differences between comparison methods affect the accuracy of ranker comparisons. In the second experiment, we use these first results to select a small subset of ranker pairs with which to study convergence characteristics in more detail. In the third, we use the same subset of rankers to examine the effect of noise in click data.

All experiments employ the same experimental setup that simulates user interactions. This setup is described in the next section (§5.1). After this, we specify our data (§5.2) and runs (§5.3).

[2]To further reduce computation, probabilities for documents below a certain rank can be set to a small constant, trading a small loss in sensitivity for gains in speed.

5.1 Experimental setup

Our experiments are based on a recently developed experimental setup that simulates user interactions based on datasets with explicit relevance judgments and user click models [12]. This simulator fulfills the assumption made by all comparison methods that more relevant documents are more likely to be clicked. Simulating user interactions makes it possible to study the comparison methods at different levels of noise in user clicks. In addition, the experiments can be repeated for many comparisons.

The simulation takes as input a dataset with queries and explicit relevance judgments and simulates user interactions as follows. It simulates a user submitting a query by randomly sampling from the set of available queries (with replacement).[3] On the system side, the interleaved comparison method observes the sampled query and produces an interleaved result list that is presented to the user (sent back to the simulator). User clicks are then simulated using a click model and the relevance judgments provided with the dataset.

The click model simulates user interactions according to the Dependent Click Model (DCM) [8, 9], an extension of the cascade model [6]. According to this model, users traverse result lists from top to bottom. For each document they encounter, they decide whether the document representation is promising enough to warrant a click (e.g., based on the title and document snippets). If, after clicking on a document, the users' information need is satisfied (likely if the document was relevant), then they stop browsing the result list. Otherwise, they continue examining the result list (likely if the document was not relevant).

We implement the click model following [12], with the difference that we define click and stop probabilities based on graded relevance assessments instead of binary ones. Simulated users examine the result list starting at the top. For each document they encounter, they decide whether to click or not, with probability $P(c|R(d))$, i.e., the probability of a click given the relevance level of the examined document $R(d)$. While the relevance level is not known to the user before clicking, we assume that its presentation in the result list is informative enough to make relevant documents more likely to be clicked. The probability that a user is satisfied with the obtained results and stops after clicking a document with relevance level $R(d)$ is modeled by $P(s|R(d))$.

We instantiate the click model in two ways (for 5 relevance lev-

[3]In the absence of information about the original distribution of the queries in the data set, we sample uniformly. An advantage of doing so is that frequent queries cannot dominate performance. Different query distributions may affect convergence speed, but not the qualitative differences between the comparison methods we evaluate.

Table 2: Instantiations of the click model used in our experiments.

perfect model					
$R(d)$	0	1	2	3	4
$P(c\|R(d))$	0.00	0.20	0.40	0.80	1.00
$P(s\|R(d))$	0.00	0.00	0.00	0.00	0.00
realistic model					
$R(d)$	0	1	2	3	4
$P(c\|R(d))$	0.05	0.10	0.20	0.40	0.80
$P(s\|R(d))$	0.00	0.20	0.40	0.60	0.80

els) to study the behavior of comparison methods under *perfect* and *realistic* conditions. The perfect click model is used to examine the behavior of comparison methods in an ideal setting. The stop probabilities are 0 for all relevance levels, i.e., the user continues to examine documents regardless of the relevance of previously encountered documents. Therefore, all documents in the top 10 of each result list are examined. The click probabilities are defined such that highly relevant documents ($R(d) = 4$) are always clicked, and that non-relevant documents ($R(d) = 0$) are never clicked. Click probabilities for relevance levels between these extremes decrease monotonically. The realistic click model was designed to approximate noisier click behavior. Here, highly relevant documents are most likely to be clicked ($P(c|R(d) = 4) = 0.8$), and click probabilities decrease for lower relevance levels. Similarly, stop probabilities for this model are highest after seeing a highly relevant document and decrease for lower probabilities. Besides the different levels of noise, the two models differ in the overall expected number of clicks, which is lower for the realistic click model because the click probabilities are consistently lower and the stop probabilities are higher. Table 2 lists the probabilities defined for the two models.

The click models instantiated here are appropriate for our evaluation of comparison methods, as they satisfy the assumptions made by all these methods. In preliminary experiments, we found that the level of noise introduced by the realistic click model is of the same order of magnitude as observed in previous work [21]. In addition, we observed qualitatively similar results for all other instantiations we tested. Finer details, such as click behavior that takes relations between documents into account are not captured. Therefore, extending our evaluation to include real-life experiments is important future work.

5.2 Data

All experiments make use of the MSLR-WEB30k Microsoft learning to rank (MSLR) data set.[4] This data set consists of approximately $30,000$ queries with 136 precomputed document features and relevance judgments. Relevance judgments are provided on a 5-point scale. Unless stated otherwise, our experiments use the training set of fold 1. This set contains $18,919$ queries, with an average of 9.6 judged documents per query.

To generate rankers to compare in our experiments, we make use of the features provided for the documents of the data set. Specifically, we take each feature to be a ranker, and formulate the task of the comparison method as identifying the ranking features that are expected to give the best performance. In our first set of experiments, we exhaustively compare all distinct ranker pairs that can be generated in this way. For the second and third sets, we select a small subset that appears most suitable for more detailed analysis.

[4]Downloaded from http://research.microsoft.com/en-us/projects/mslr/default.aspx.

The experiments are designed to evaluate the comparison methods' ability to identify the better of two rankers based on (simulated) user clicks. To set the ground truth for which ranker is better, we use NDCG based on the explicit relevance judgments provided with the data set, following previous work [21]. We use the complete set of documents provided with the data set to determine NDCG for the ground truth, i.e., no cut-off is used. Note that comparisons between rankers using implicit feedback depend not only on the true difference in NDCG between two rankers but also on the magnitude of the difference per query and the number of affected queries.

5.3 Runs

All our experiments evaluate the same four interleaved comparison methods. As baselines, we use three existing methods (cf., §2):

- **balanced interleave**, as described in [23].
- **team draft**, as introduced in [23]; as in the original, clicks on top results are ignored when the documents are placed in identical order by both rankers.
- **document constraint**, introduced in [10], as described in §2.

In addition, we evaluate the experimental condition:

- **marginalized probabilities**: our model using the probabilistic interleave process (cf. §4.1) and the probabilistic comparisons defined in Eq. 4 (cf. §4.3).

Below we describe the three sets of experiments designed to assess interleaved comparison methods in terms of accuracy, convergence behavior, and robustness to noise.

Accuracy. The first experiment assesses the accuracy of interleaved comparison methods when evaluated on a large set of ranker pairs. We construct a ranker for each of the 136 individual features provided with the MSLR data set. We then evaluate the interleaved comparison methods on all $9,180$ possible distinct pairs that can be obtained from these rankers. We evaluate the methods on this large set of ranker pairs using the perfect click model and a fixed subset of the data consisting of $1,000$ queries. We then compare the outcome predicted by the interleaved comparison methods to the true NDCG of the rankers, computed on the same set of queries. We report on *accuracy* after $1,000$ queries: the percentage of pairs for which a comparison method correctly identified the better ranker.

Convergence. Based on the findings of the first experiment, we select a small subset of ranker pairs with which to investigate the methods' behavior in more detail. These experiments were designed to follow the setup used in [21] as closely as possible, with the difference that we use simulated interactions with the perfect click model instead of real-life search engine traffic, as discussed in §5.1. For each pair of rankers, each comparison method was run on randomly sampled query sets of size 1 to $10,000$; queries for which an interleaved list did not obtain any clicks were ignored. Each experiment was repeated $1,000$ times. We evaluate each method by computing, for each sampled query set size, the fraction of these $1,000$ repetitions for which it detected a preference for the correct ranker. We compute binomial confidence intervals on this fraction using Wilson score intervals at the 95% level [3].

Noise. The third set of experiments assesses the robustness of interleaved comparison methods to noise in click feedback. The setup is identical to the previous one except that the *realistic* click model is used instead.

6. RESULTS AND DISCUSSION

In this section we present and discuss the results of our experiments. Results from the exhaustive experiment, designed to assess the accuracy of interleaved comparison methods are presented first (§6.1). Based on the results of that experiment we select a subset of rankers to further investigate the convergence of these methods (§6.2), and their robustness to noise in click feedback (§6.3).

6.1 Accuracy

Table 3 gives an overview of the accuracy achieved by the four interleaved comparison methods in our first set of experiments. The highest accuracy is achieved by our probabilistic marginalization method. It correctly identifies the better ranker for 91.4% of the pairs. This constitutes a 1.7% increase (or 143 additional pairs) in accuracy over the second-best run, team draft. The team draft method achieves an accuracy of 89.8%, followed by balanced interleave at 88.1%. The accuracy achieved by the document constraint method is substantially lower (85.7%).

Table 3: Accuracy for the comparison methods on ranker pairs constructed from individual features (after 1,000 queries).

run	*accuracy*
balanced interleave	0.881
team draft	0.898
document constraint	0.857
marginalized probabilities	**0.914**

To gain further insight into the performance characteristics of each method, we analyzed what type of ranker pairs each method judged correctly or incorrectly. The balanced interleave, team draft, and marginalized probabilities methods correctly identify the better ranker in all pairs with an NDCG difference of at least 0.05. This is in line with [21], which found that the team draft method can be used to reliably compare ranker pairs with a difference in NDCG of 0.05 within approximately 1,000 queries. The document constraint method made mistakes on much more distant ranker pairs, with an NDCG difference of up to 0.12.

We further find that the document constraint method makes a number of systematic errors, meaning that an unexpectedly large number of pairs involving a particular ranker are judged incorrectly. For example, a large number of comparisons involving the ranker constructed from one of the best features, query-click count (feature id 134 in the MSLR data set), are incorrect. The ranked lists that are generated in these cases differ substantially, so that there is little overlap. The documents ranked highly by the better ranker contain many more relevant documents, so more constraints involving these documents are inferred from clicks. As most of these documents are not included in the top ranks of the worse ranker, the inferred constraints are not counted as being violated by this ranker, while they do count against the better ranker.

For the balanced interleave method, we also observe systematic errors, but to a much lesser degree. These errors typically involve pairs of rankers that are very similar, or that are likely related, such as the minimum term frequency and the minimum tf-idf (feature id 27 and 77, respectively). For the team draft and marginalized probabilities methods, we found no such systematic bias. Errors appear to be randomly distributed over ranker pairs.

The results of our first set of experiments indicate that the marginalized probabilities method is more accurate than other methods. On a large set of rankers it was able to infer the correct relation between rankers the most often. This method, as well as balanced interleave and team draft, are able to reliably compare rankers with a difference in NDCG of at least 0.05 after 1,000 queries. For the document constraint method, we found systematic errors even when the difference in NDCG between two rankers was very large.

Table 4: Ranker pairs used for the second set of experiments.

pair	*ranking features (feature id)*	*NDCG*
	"easy" ranker pairs	
1	term frequency of the document url, normalized by mean stream length (64)	0.301
	mean stream length of the document url (14)	0.201
2	bm25 score on the document title (108)	0.306
	maximum tf-idf on the url (84)	0.256
	"problem" ranker pairs	
3	click-count for a query-url pair of a search engine over a given period (134)	0.341
	boolean retrieval score on the document body (96)	0.219
4	boolean retrieval score on anchor text (97)	0.303
	bm25 score on the document body (106)	0.253
5	minimum tf-idf on anchor text (77)	0.262
	number of covered query terms (1)	0.231

6.2 Convergence

Based on the outcomes of the first set of experiments, we selected a small set of ranker pairs for more in-depth analysis. We used these rankers to run a second set of experiments with a large number of randomly sampled queries and many repetitions, to distinguish performance characteristics due to chance from more systematic issues.

The ranker pairs used for this second set of experiments are shown in Table 4. For each pair, the table includes a description of the two ranking features, the corresponding feature id in the MSLR data set, and the NDCG that is obtained on the whole data set if the feature is used for ranking documents for each query.

The pairs of rankers were selected as follows. First, we selected two pairs of rankers for which all methods had correctly identified the better ranker. These "easy" pairs have an NDCG difference of 0.1 (pair 1) and 0.05 (pair 2) percentage points, respectively. The second pair is on the threshold at which three of the methods achieved perfect accuracy within 1,000 queries, and corresponds to the "major" experiments in [21].

Second, we selected a set of "problem" pairs, on which one or more of the comparison methods had made mistakes. For each method, we selected the incorrectly judged pair of rankers with the highest difference in NDCG. Because pairs with a large difference in NDCG are typically easier to distinguish, these pairs are most useful for diagnosis. If the method again comes to the wrong conclusion after evaluation with a much larger set of queries and repetitions, then we can conclude that the mistake is systematic and that similar errors will likely occur for more difficult ranker pairs with smaller differences in NDCG.

The problematic pair selected for the document constraint method is pair 3. This method made many mistakes involving the click-count feature (134), and we select the pair with the highest NDCG difference (0.12). Pair 4 is the pair that was judged incorrectly by the team draft and marginalized probabilities methods with the highest difference in NDCG (0.05). As these runs did not appear to have systematic difficulties, we assumed that a pair that had been judged wrong by both methods would be most likely to point to a systematic problem. The balanced interleave method made systematic mistakes on pairs with relatively small differences, of which the selected pair 5 is the one with the biggest difference in NDCG (0.03).

Figures 6 to 9 show the results of our second set of experiments, using the perfect click model. The figures show the fraction of repetitions in which the hypothesized better ranker won the comparison (on the y-axis) vs. the number of sampled queries (on the x-axis). A comparison method that performs well will converge to 1 quickly, meaning that it detects a preference for the correct ranker after observing a relatively small number of samples.

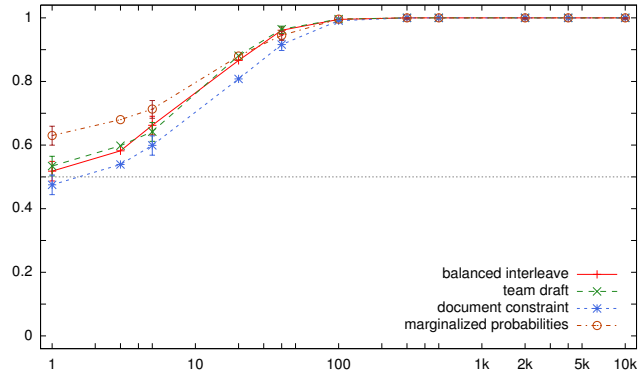

Figure 6: Results, pair 1, perfect click model (see text for results for pair 2).

Figure 6 shows the results obtained when comparing pair 1 using (simulated) user clicks. We observe that for "easy" pairs, all methods converge to the correct decision quickly, in this case after approximately 100 observed queries. Results for pair 2 are qualitatively similar to those obtained for pair 1, so we only include one of the figures due to space limitations. Convergence on pair 2 is slower, as expected. All methods converge within $1,000$ observed queries, a result that is in line with [21].

To further investigate what constitutes "easy" ranker pairs, we analyzed how NDCG differences are distributed over queries. For pair 1 and 2 we found that these differences are distributed over a large portion of queries. Thus, a consistent NDCG difference on a large number of queries appears to make the detection of such differences using interleaving techniques easiest.

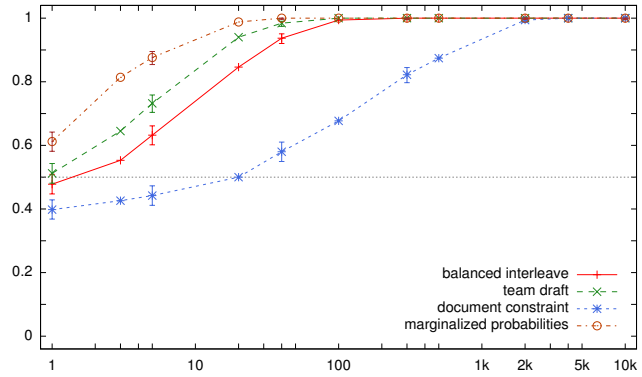

Figure 7: Results, pair 3, perfect click model.

Figure 7 shows the results obtained for pair 3, which was selected as a problematic pair for the document constraint method. Indeed, this method is slowest to converge. However, it does identify the hypothesized better ranker correctly, and converges within approximately $2,000$ observed samples (half as fast as other methods). Marginalized probabilities converges significantly faster than all other methods, after only 50 observed samples.

To analyze this performance difference, we again examine the NDCG differences between rankers and how they are distributed over queries. We find that the relatively large difference in NDCG is distributed over a relatively small number of queries. Thus, there is a relatively small set of queries for which the first ranker performs dramatically better than the second and a large set of queries for which the difference is negligible, or where the second ranker is preferred. In such situations, the magnitude of the differences between rankers can play a critical role. Our method's use of softmax functions and marginalization enables it to more accurately assess these magnitudes, leading to better, faster comparisons.

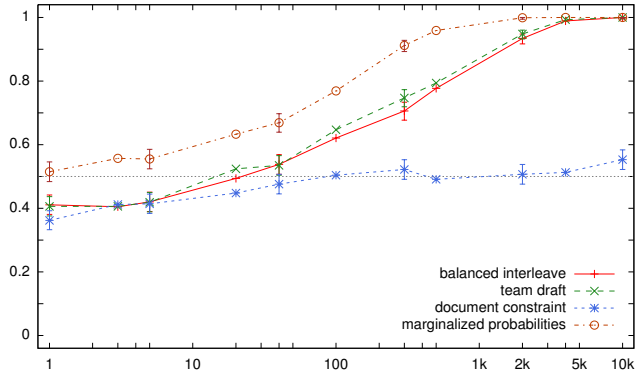

Figure 8: Results, pair 4, perfect click model.

Results for pair 4 are presented in Figure 8. This pair of rankers was selected as problematic for the team draft and marginalized probabilities methods. However, it appears that the selected pair is simply more difficult than previous ones and that this difficulty is not specific to these two methods. Though convergence on this pair is much slower than on those discussed previously, the marginalized probabilities method still converges much faster than other methods ($\approx 2,000$ impressions, vs. $4,000$ for the second-fastest team draft method). Note that the document constraint method has still not converged to a decision after $10,000$ impressions.

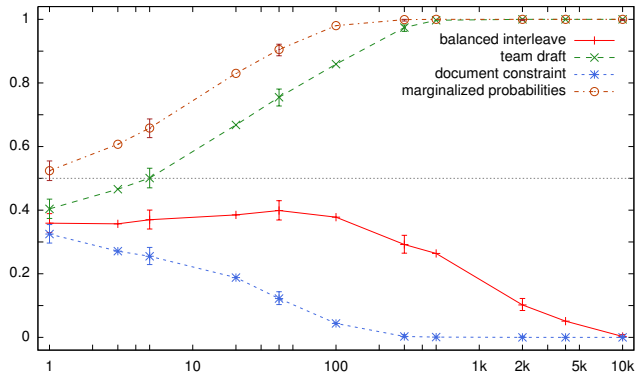

Figure 9: Results, pair 5, perfect click model.

Results for the last pair of rankers we examine, pair 5, are shown in Figure 9. This pair was selected as problematic for the balanced interleave method. The NDCG difference of this pair is smallest (0.03), as is the number of affected queries. The team draft and marginalized probabilities methods quickly identify the correct ranker, with the marginalized probabilities method converging much faster than the team draft method. The other two methods are unable to identify the better ranker for this pair. For this pair, the interleaved lists tend to be very similar to the worse ranker, cre-

ating bias towards that ranker. As a result, the balanced interleave and document constraint methods converge to the wrong decision.

Overall, results for all the pairs considered in this section show a similar pattern. The marginalized probabilities method performs best overall, identifying the better rankers correctly for all pairs, and converging to the correct solution quickly, often substantially and significantly faster than all other methods. The team draft method performs second best, identifying the better ranker correctly in all cases and typically converging faster than the balanced interleave and document constraint methods, which select the wrong ranker in one case. Of course, there may exist pairs of rankers for which the document constraint and balanced interleave methods converge faster than team draft and probabilistic marginalization due to bias towards the ranker that is in fact better. However, there is in general no guarantee that these methods will select the better ranker, no matter how large the number of observed queries.

6.3 Noise

In the previous section, we studied the performance of interleaved comparison methods under simulated perfect conditions, in which the simulated user examined all top 10 documents, clicked on all highly relevant and no non-relevant documents (with click probabilities decreasing for in-between relevance levels; cf. §5.1). Here, we use a more realistic user model to study the influence of noise on the performance of the comparison methods. Our *realistic* click model defines click probabilities such that there is a 0.8 probability of highly relevant documents being clicked, with click probabilities cut in half for each subsequently lower relevance level. Stop probabilities increase with the relevance level of previously seen documents. Hence, the chance that clicks prefer a less relevant document is higher than with the perfect model and documents ranked towards the top of the result list are more likely to be clicked. Ideally, noise should slow the convergence of interleaved comparison methods but not alter the final outcome.

For the third set of experiments, we omit graphs for pairs 1, 2, and 5, as the results are very similar to those presented in the previous section. For pairs 1 and 2, all methods identify the better ranker correctly but converge more slowly than under perfect conditions. For pair 1, the correct ranker is identified after between 300 impressions (marginalized probabilities) and 500 impressions (document constraint), respectively; for pair 2 this occurs after approximately 2,000 impressions. For pair 2, only the marginalized probabilities and team draft methods identify the correct rankers (after approximately 2,000 impressions).

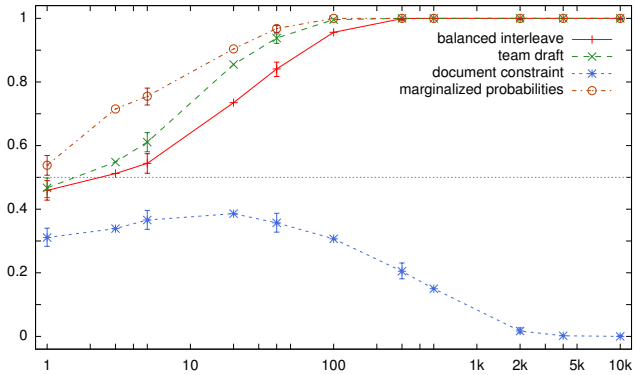

Figure 10: Results, pair 3, realistic click model.

Interesting performance differences can be seen for pairs 3 and 4. For pair 3 (Figure 10), noise affects the outcome for the document constraint method to such an extent that the method is unable to identify the better ranker. The reason for this change in outcome is that the realistic user model results in noisier clicks that are more concentrated towards the top of the result list. Documents contributed by the better ranker are generally more relevant and clicked more often in expectation. Clicks on documents other than the top-ranked one will result in constraints that are more often violated by the better ranker, while fewer clicks on documents contributed by the worse ranker result in fewer violated constraints for that ranker. This result suggests that the document constraint method is not appropriate for use with noisy click feedback, especially when the true quality difference between rankers is very large.

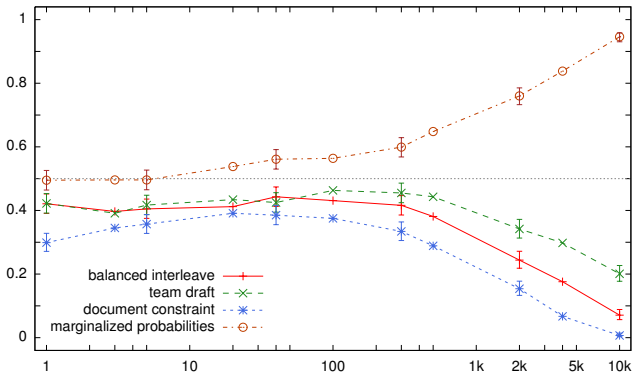

Figure 11: Results, pair 4, realistic click model.

Figure 11 shows the noisy results obtained with pair 4. Only the marginalized probabilities method correctly identifies the better ranker and it converges more slowly than under perfect conditions. None of the baseline methods identifies the correct ranker. For the team draft method, which preferred the predicted better ranker in the perfect setting, the influence of more realistic stop probabilities plays a role. Previously, the simulated user examined all results in the top 10 interleaved list, so relevant documents at similar ranks had the same chance of being clicked. Now, the simulated user may stop browsing the result list after clicking on the link to a document, creating a disadvantage for one of the rankers, especially in cases where rankers place relevant documents at similar ranks (which is particularly likely in cases where the rankers produce similar result lists). Here, this phenomenon creates a systematic disadvantage for the true better ranker. This demonstrates that an insensitivity to some changes in rankings can result in the team-draft method inferring a preference for the wrong ranker.

Our evaluation of interleaved comparison methods with noisy and less exhaustive user clicks shows that our marginalized probabilities method is robust to such changes in click behavior. In all cases, the method was able to correctly identify the better ranker. As predicted, noise slowed convergence. For all the baseline methods, the addition of noise caused them to infer a preference for the wrong ranker more often than under perfect conditions.

7. CONCLUSION

Methods for evaluating rankers using implicit feedback are promising because implicit feedback can be collected directly while users interact with a search engine, which reduces cost and allows tuning to the true preferences of users. Previous work introduced the *balanced interleave*, *document constraint*, and *team draft* methods for comparing rankers using click feedback based on interleaving lists. The *team draft* method has previously been shown to reliably detect small differences between rankers in practice.

The first contribution of our paper is an analysis of interleaved

comparison methods. We formulated criteria in terms of bias and sensitivity and analyzed how well existing methods fulfill them. In addition to the previously known bias of the balanced interleave method, we identified a similar bias affecting the document constraint method, and a lack of sensitivity to specific differences in ranking for the team draft method.

Our second contribution is a new, probabilistic interleaved comparison method that is both unbiased and sensitive to differences between rankings. Central to the method is the use of softmax functions, which assign to each document a non-zero probability of being placed in the interleaved result list. As a result, click assignment to rankers is smoothed, yielding credit assignment that accurately reflects the magnitude of differences between rankers. Furthermore, we showed how the method can be made more efficient by marginalizing over all possible assignments for an observed interleaved list.

We validated the results of our analysis and our proposed method using experiments based on a recently developed simulation framework. The probabilistic method for comparing rankers using click data is more accurate than any of the previous methods. In particular, on pairs of rankers that are difficult for other methods to compare, e.g., because the rankers perform very differently on only a few queries, our method can correctly identify the better ranker, with substantially and significantly fewer observed queries. In addition, the method is more robust to noise than previous methods.

Our simulation setup allowed us to undertake many comparisons under different conditions (e.g., varying noise). However, as with any simulation, these experiments cannot replace experiments with real users. Therefore, in the future we aim to evaluate our method in a real-life setting. In addition, with more reliable and faster convergence, our approach can pave the way for online learning to rank methods that require many comparisons. Applying the method in such settings is another interesting direction for future research.

Acknowledgements

This research was partially supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430, the PROMISE Network of Excellence co-funded by the 7th Framework Programme of the European Commission, grant agreement no. 258191, the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.061.814, 612.061.-815, 640.004.802, 380-70-011, the Center for Creation, Content and Technology (CCCT), the Hyperlocal Service Platform project funded by the Service Innovation & ICT program, the WAHSP project funded by the CLARIN-nl program, and under COMMIT project Infiniti.

8. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06*, pages 19–26, 2006.

[2] P. Bailey, N. Craswell, I. Soboroff, P. Thomas, A. P. de Vries, and E. Yilmaz. Relevance assessment: are judges exchangeable and does it matter. In *SIGIR '08*, pages 667–674, 2008.

[3] L. D. Brown, T. T. Cai, and A. DasGupta. Interval estimation for a binomial proportion. *Statistical Science*, 16(2):pp. 101–117, 2001.

[4] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *SIGIR '09*, pages 1–10, 2009.

[5] M. Claypool, P. Le, M. Wased, and D. Brown. Implicit interest indicators. In *IUI'01*, IUI '01, pages 33–40, 2001.

[6] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *WSDM '08*, pages 87–94, 2008.

[7] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM TOIS*, 23(2):147–168, 2005.

[8] F. Guo, L. Li, and C. Faloutsos. Tailoring click models to user goals. In *WSCD '09*, pages 88–92, 2009.

[9] F. Guo, C. Liu, and Y. M. Wang. Efficient multiple-click models in web search. In *WSDM '09*, pages 124–131, 2009.

[10] J. He, C. Zhai, and X. Li. Evaluation of methods for relative comparison of retrieval systems based on clickthroughs. In *CIKM '09*, pages 2029–2032, 2009.

[11] K. Hofmann, B. Huurnink, M. Bron, and M. de Rijke. Comparing click-through data to purchase decisions for retrieval evaluation. In *SIGIR '10*, pages 761–762, 2010.

[12] K. Hofmann, S. Whiteson, and M. de Rijke. Balancing exploration and exploitation in learning to rank online. In *ECIR'11*, pages 251–263, 2011.

[13] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4): 422–446, 2002.

[14] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02*, pages 133–142, 2002.

[15] T. Joachims. Evaluating retrieval performance using clickthrough data. In J. Franke, G. Nakhaeizadeh, and I. Renz, editors, *Text Mining*, pages 79–96. Springer Verlag, 2003.

[16] S. Jung, J. L. Herlocker, and J. Webster. Click data as implicit relevance feedback in web search. *IPM*, 43(3): 791 – 807, 2007. Special Issue on Heterogeneous and Distributed IR.

[17] D. Kelly. Implicit feedback: Using behavior to infer relevance. In *New directions in cognitive information retrieval*, pages 169–186. Springer, 2005.

[18] D. Kelly and N. Belkin. Display time as implicit feedback: understanding task effects. In *SIGIR '04*, pages 377–384, 2004.

[19] J. Langford, A. Strehl, and J. Wortman. Exploration scavenging. In *ICML '08*, pages 528–535, 2008.

[20] R. Lippmann. Pattern classification using neural networks. *Communications Magazine, IEEE*, 27(11): 47–50, 2002.

[21] F. Radlinski and N. Craswell. Comparing the sensitivity of information retrieval metrics. In *SIGIR '10*, pages 667–674, 2010.

[22] F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. In *ICML '08*, pages 784–791, 2008.

[23] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? In *CIKM '08*, pages 43–52, 2008.

[24] F. Scholer, M. Shokouhi, B. Billerbeck, and A. Turpin. Using clicks as implicit judgments: expectations versus observations. In *ECIR'08*, pages 28–39, 2008.

[25] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *SIGIR '05*, pages 43–50, 2005.

[26] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1998.

[27] R. White, I. Ruthven, J. Jose, and C. Rijsbergen. Evaluating implicit feedback models using searcher simulations. *ACM TOIS*, 23(3):325–361, 2005.