Estimating Position Bias without Intrusive Interventions

Aman Agarwal
Cornell University
Ithaca, NY
aa2398@cornell.edu

Ivan Zaitsev
Cornell University
Ithaca, NY
iz44@cornell.edu

Xuanhui Wang, Cheng Li, Marc Najork
Google Inc.
Mountain View, CA
{xuanhui,chgli,najork}@google.com

Thorsten Joachims
Cornell University
Ithaca, NY
tj@cs.cornell.edu

ABSTRACT

Presentation bias is one of the key challenges when learning from implicit feedback in search engines, as it confounds the relevance signal. While it was recently shown how counterfactual learning-to-rank (LTR) approaches [18] can provably overcome presentation bias when observation propensities are known, it remains to show how to effectively estimate these propensities. In this paper, we propose the first method for producing consistent propensity estimates without manual relevance judgments, disruptive interventions, or restrictive relevance modeling assumptions. First, we show how to harvest a specific type of intervention data from historic feedback logs of multiple different ranking functions, and show that this data is sufficient for consistent propensity estimation in the position-based model. Second, we propose a new extremum estimator that makes effective use of this data. In an empirical evaluation, we find that the new estimator provides superior propensity estimates in two real-world systems – Arxiv Full-text Search and Google Drive Search. Beyond these two points, we find that the method is robust to a wide range of settings in simulation studies.

KEYWORDS

Unbiased learning-to-rank, counterfactual inference, propensity estimation

1 INTRODUCTION

In most information retrieval (IR) applications (e.g., personal search, scholarly search, product search), implicit user feedback (e.g. clicks, dwell time, purchases) is routinely logged and constitutes an abundant source of training data for learning-to-rank (LTR). However, implicit feedback suffers from presentation biases, which can make its naive use as training data highly misleading [16]. In particular, the position at which a result is displayed introduces a strong bias, since higher-ranked results are more likely to be discovered by the user than lower-ranked ones.

It was recently shown that counterfactual inference methods provide a provably unbiased and consistent approach to LTR despite biased data [18]. The key prerequisite for counterfactual LTR is knowledge of the propensity of obtaining a particular feedback signal, which enables unbiased Empirical Risk Minimization (ERM) via Inverse Propensity Score (IPS) weighting. This makes getting accurate propensity estimates a crucial prerequisite for effective unbiased LTR.

In this paper, we propose the first approach for producing consistent propensity estimates without manual relevance judgments, disruptive interventions, or restrictive relevance-modeling assumptions. We focus on propensity estimation under the Position-Based Propensity Model (PBM), where all existing propensity estimation methods have substantial drawbacks. In particular, the conventional estimator for the PBM takes a generative approach [7] and requires individual queries to repeat many times. This is unrealistic for most ranking applications. To avoid this requirement of repeating queries, Wang et al. [29] included a relevance model that is jointly estimated via an Expectation-Maximization (EM) procedure [9]. Unfortunately, defining an accurate relevance model is just as difficult as the learning-to-rank problem itself, and a misspecified relevance model can lead to biased propensity estimates. An alternative to generative click modeling on observational data are estimators that rely on specific randomized interventions – for example, randomly swapping the result at rank 1 to any rank k [18]. While this provides provably consistent propensity estimates for the PBM, it degrades retrieval performance and user experience [29] and is thus costly. In addition, we find in the following that swap-interventions are statistically rather inefficient. The approach we propose in this paper overcomes the disadvantages of the existing methods, as it does not require repeated queries, a relevance model, or additional interventions.

The key idea behind our estimation technique is to exploit data from a natural intervention that is readily available in virtually any operational system – namely that we have implicit feedback data from more than one ranking function. We call this Intervention Harvesting. Since click behavior depends jointly on examination and relevance, we show how to exploit this intervention to control for any difference in overall relevance of results at different positions under the PBM. This makes our approach fundamentally

interventional and it is thus consistent analogous to using explicit swap interventions under mild assumptions. However, by leveraging existing data that is readily available in most systems, it does not require additional online interventions and the resulting decrease in user experience. To make efficient use of the interventional data we harvest, we propose a specific extremum estimator – called AllPairs – that combines all available data for improved statistical efficiency without the need for a relevance model that could introduce bias. We find that this estimator works well even if the rankers we harvest interventions from are quite similar in their result placements and overall performance, and that it is able to recover the relative propensities globally even if most of the changes in rank are small.

2 RELATED WORK

Implicit user feedback (e.g. clicks and dwell time) has been widely used to improve search quality. In order to fully utilize the implicit feedback signals for LTR algorithms, various types of bias have to be handled [18], e.g., position bias [15], presentation bias [30], and trust bias [15, 22]. To address this challenge, a large amount of research has been devoted to extracting more accurate signals from click data. For example, some heuristic methods have been proposed to address the position bias by utilizing the pairwise preferences between clicked and skipped documents [14–16]. Though these methods have been found to provide more accurate relevance assessments, their data is still biased. For example, click vs. skip preference tend to reverse the presented order when used for learning [14] due to their sampling bias.

Recently, Joachims et al. [18] presented a counterfactual inference framework, which provides a provably unbiased and consistent approach to LTR even with biased feedback data. It requires knowledge of the propensity of obtaining a particular feedback signal, based on which an Inverse Propensity Scoring (IPS) approach can be applied. IPS was developed in causal inference [24] and is a widely accepted technique for handling sampling bias. It has been employed in unbiased evaluation and learning [1, 10, 20, 21, 25, 26]. The common assumption in most of these studies is that the propensities are under the system's control and are thus known. In the unbiased LTR setting, however, propensities arise from user behavior and thus need to be estimated.

The problem of propensity estimation for LTR was already addressed in other works. Wang et al. [28] and Joachims et al. [18] proposed to estimate propensity via randomization experiments. Carterette and Chandar [4] extend the counterfactual inference framework [18] by considering the case of evaluating new rankers that can retrieve previously unseen documents. Their methods still rely on interventions, though they make the interventions minimally invasive. To avoid intrusive interventions that degrade the user search experience, Wang et al. [29] proposed a regression-based EM algorithm to estimate position bias without interventions. Similarly, Ai et al. [3] presented a framework to jointly learn an unbiased ranker and an unbiased propensity model from biased click data. Both works couple relevance estimation with propensity estimation, which introduces the drawback of potential bias when the relevance model is misspecified. Our work differs from these by

controlling for relevance explicitly and without a relevance model, while still avoiding intrusive interventions.

Another line of research employs click models [7] to infer relevance judgments from click logs. Training is typically performed via generative maximum likelihood under specific modeling assumptions about user behavior. There are two classic clicks models: the position-based model (PBM) [23] and the Cascade model [8]. Based on these two models, more advanced models have been developed, including UBM [11], DBN [6], and CCM [12]. Our propensity model is based on the PBM. However, we do not use it as a generative model to infer relevance, but instead use interventional techniques to infer propensities even without repeat queries.

3 HARVESTING INTERVENTIONS

We approach the propensity estimation task by harvesting implicit interventions from already logged data. In particular, we make use of the fact that we typically have data from multiple historic ranking functions, and we will identify the conditions under which these provide unconfounded intervention data. We focus on the Position-Based Propensity Model (PBM), which we review before defining interventional sets and analyzing their properties.

3.1 Position-Based Propensity Model

The position-based model recognizes that higher-ranked results are more likely to be considered (i.e. discovered and viewed) by the user than results further down the ranking. Suppose that for a particular query q, result d is displayed at position k. Let C be the random variable corresponding to a user clicking on d, and let E be the random variable denoting whether the user examines d. In our notation, q represents all information about the users, the query context, and which documents the user considers relevant or not. This mean we can denote the relevance of an individual document as a non-random function $rel(q, d)$ of q, where $rel(q, d) = 1$ and $rel(q, d) = 0$ indicates relevant and non-relevant respectively. Then according to the Position-Based Propensity Model (PBM) [7],

$$\Pr(C = 1|q, d, k) = \Pr(E = 1|k) \, rel(q, d)$$
$$= p_k \, rel(q, d).$$

In this model, the examination probability $p_k := \Pr(E = 1|k)$ depends only on the position, and it is identical to the observation propensity [18]. For learning, it is sufficient to estimate relative propensities p_k/p_1 for each k [18], since multiplicative scaling does not change the training objective of counterfactual learning methods (e.g. [2, 3, 17, 18, 26, 27]). Estimating these relative propensities is the goal of this paper.

Note that one can train multiple PBM models to account for changes in the propensity curve due to context. In this way, the expressiveness of the PBM model can be substantially extended. For example, one can train separate PBM models for navigational vs. informational queries simply by partitioning the data. While training multiple such models is prohibitively expensive when intrusive interventions are required, the intervention harvesting approach we describe below makes training such contextual PBMs feasible since it does not require costly data, is both statistically and computationally efficient, and does not have any parameters that require manual tuning.

3.2 Controlling for Relevance through Swap Interventions

The key problem in both propensity estimation and unbiased learning to rank is that we only observe clicks C, but we never get to observe E (whether a user examined a result) and $\text{rel}(q, d)$ (whether the user found d relevant for q) individually. This makes it difficult to attribute the lack of a click to a lack of examination or a lack of relevance. A key idea for overcoming this dilemma without explicit relevance judgments or cumbersome instrumentation (e.g. eye tracking) was proposed in [18], namely to control for relevance through randomized interventions.

The intervention proposed in [18] is to randomly swap the result in position 1 with the result in position k. We call these $Swap(1, k)$-interventions. Such swaps provide a completely randomized experiment [13], meaning that the assignment of the document to a position does not depend on relevance or any covariates (e.g. abstract length, document language). Under these swap interventions, we now get to observe how many clicks position 1 results get when they stay in position 1 vs. when they get swapped into position k. Since the expected relevance in either condition (i.e. swap vs. not swapped) is the same, any change in clickthrough rate must be proportional to a drop in examination.

More formally, we model user queries as sampled i.i.d. $q \sim \Pr(Q)$. Whenever a query is sampled, the ranker $f(q)$ sorts the candidate results d for the query and we apply the randomized swap intervention between positions 1 and some fixed k before the ranking is displayed to the user. As a precursor to the later exposition, suppose that the random swap occurs with a fixed probability p (not necessarily 0.5), yielding the logged datasets $D_1^{1,k} = (q_1^i, d_1^i, C_1^i)^{n_1}$ (result stayed in position 1) and $D_k^{1,k} = (q_k^j, d_k^j, C_k^j)^{n_2}$ (result was swapped into position k) of sizes n_1 and n_2 respectively. Here C_1^i denotes whether the document d_1^i placed at position 1 was clicked or not, and similarly for C_k^j. Denote with $\hat{c}_1^{1,k} = \frac{1}{n_1} \sum C_1^i$ the rate of clicks that documents get when they remain in position 1 and let $\hat{c}_k^{1,k} = \frac{1}{n_2} \sum C_k^j$ be the rate of clicks when they get swapped to position k. Then, under the PBM, the relative propensity is equal to the ratio of expected click rates:

$$
\begin{aligned}
\frac{p_k}{p_1} &= \frac{p_k E_{D_k^{1,k}} \left[\sum \text{rel}(q_k^j, d_k^j) / n_2 \right]}{p_1 E_{D_1^{1,k}} \left[\sum \text{rel}(q_1^i, d_1^i) / n_1 \right]} \\
&= \frac{E_{D_k^{1,k}} \left[\sum p_k \, \text{rel}(q_k^j, d_k^j) / n_2 \right]}{E_{D_1^{1,k}} \left[\sum p_1 \, \text{rel}(q_1^i, d_1^i) / n_1 \right]} \\
&= \frac{E_{D_k^{1,k}} \left[\sum \Pr(C_k^j = 1 | q_k^j, d_k^j, k) / n_2 \right]}{E_{D_1^{1,k}} \left[\sum \Pr(C_1^i = 1 | q_1^i, d_1^i, 1) / n_1 \right]} \\
&= \frac{E_{D_k^{1,k}} \left[\hat{c}_k^{1,k} \right]}{E_{D_1^{1,k}} \left[\hat{c}_1^{1,k} \right]}
\end{aligned}
$$

The first equality holds since swaps are completely at random, such that the expected average relevance under each condition is equal and their ratio is one (we need not consider n_1 and n_2 as random variables). It is thus reasonable to estimate the relative propensity

as

$$
\frac{\hat{p}_k}{\hat{p}_1} = \frac{\hat{c}_k^{1,k}}{\hat{c}_1^{1,k}},
$$

which is a statistically consistent estimate as the sample size n grows [18]. Analogously, relative propensities can be estimated between any pairs of positions k and k' with $Swap(k, k')$ interventions [29].

3.3 Interventional Sets from Multiple Rankers

A key shortcoming of the swap experiment is its impact on user experience, since retrieval performance can be degraded quite substantially. This is especially true for swaps between position 1 and k for large k, even though this particular swap experiment makes it easy to estimate propensities relative to position 1 directly. To overcome this impact on user experience, we now show how to harvest interventions from already existing data under mild assumptions, so that no additional swap experiments are needed. As part of this, we may never see a direct swap between position 1 and k, and we tackle the problem of how to aggregate many local swaps into an overall consistent estimate in Section 4.

Our key idea in harvesting swap interventions lies in the use of data from multiple historic rankers. Logs from multiple rankers are typically available in operational systems, since multiple rankers may be fielded at the same time in A/B tests, or when the production ranker is updated frequently. Consider the case where we have data from m historic rankers $F = \{f_1, ..., f_m\}$. Furthermore, a crucial condition is that the query distribution must not depend on the choice of ranker f_i (where the query q includes the user's relevance vector rel in our notation),

$$
\forall f_i : \Pr(Q | f_i) = \Pr(Q) \Rightarrow \forall q \in Q : \Pr(f_i | q) = \Pr(f_i) \quad (1)
$$

Note that the condition on the left implies that $\Pr(f_i | q) = \Pr(f_i)$ on the right by Bayes rule, which is related to exploration scavenging [19]. Intuitively, this condition avoids that different rankers f_i get different types of queries. For rankers that are compared in an A/B test, this condition is typically fulfilled by construction since the assignment of queries to rankers is randomized. For data from a sequence of production rankers, one needs to be mindful that any temporal covariate shift in the query and user distribution is acceptably small.

We denote the click log of each ranker f_i with $\mathcal{D}_i = (q_i^j, y_i^j, c_i^j)^{n_i}$, where n_i is the number of queries that f_i processed. Here $j \in [n_i]$, q_i^j is a query, $y_i^j = f_i(q_i^j)$ is the presented ranking, and c_i^j is a vector that indicates for each document click or no click. Since most retrieval systems only rerank a candidate set of documents in their final stage, we denote $\Omega(q)$ as the candidate set of results for query q. We furthermore denote the rank of candidate result d in ranking y_i^j as $\text{rk}(d | y_i^j)$, and we use $c_i^j(d) \in [0, 1]$ to denote whether result d was clicked or not.

We begin by defining *interventional sets* $S_{k, k'}$ of query-document pairs, where one ranking function $f \in F$ puts document d at position k and another ranker $f' \in F$ puts the same document d at position k' for the same query q. Let M be some fixed number of top positions for which propensity estimates are desired (e.g. $M = 10$). Then, for each two ranks $k \neq k' \in [M]$, we define the interventional

set as

$$S_{k,k'} := \{(q,d) : q \in Q, d \in \Omega(q),$$
$$\exists f, f' \; \mathrm{rk}(d|f(q)) = k \wedge \mathrm{rk}(d|f'(q)) = k'\}$$

Intuitively, the pairs in these sets are informative because they receive different treatments or interventions based on the choice of different rankers. But note that we are *not* requiring any query to occur multiple times. An interventional set merely reflects that two potential outcomes (i.e. document d either in position k or in position k' for query q) were possible. In fact, we only ever observe one factual outcome, while the other outcome remains counterfactual and unobserved.

The key insight is that for any pair of ranks (k, k'), the interventional set contains the query-document pairs (q, d) for which the rank of d was randomly assigned to either k or k' via the choice of ranking function f_i. Specifically, there are three possible outcomes depending on the choice of ranking function $f_i \in F$: (a) f_i puts d at rank k, (b) f_i puts d at rank k', or (c) f_i puts d at some other rank. In the latter case, the instance is not included in the interventional set $S_{k,k'}$, but the former two cases can be seen as the two conditions of a swap experiment between positions k and k'. In this way, we can think about these as virtual swap interventions between ranks k and k' where the randomization comes from the randomized choice of ranking function according to (1).

While this swap experiment is completely randomized under condition (1) (i.e. the choice of f_i does not depend on q), the assignment is generally not uniform (i.e. d could have a higher probability to be presented in position k than in position k'). The weights

$$w(q, d, k) := \sum_{i=1}^{m} n_i \mathbb{1}[\mathrm{rk}(d|f_i(q)) = k].$$

are then used to account for this non-uniformity. Specifically, the weight $w(q, d, k)$ reflects how often document d is ranked at position k given that we have employed each ranking function f_i exactly n_i times. Therefore, the probability of assigning d to k as opposed to k' for query q can be estimated as

$$P(rank = k | rank = k \text{ or } rank = k', q, d) = \frac{w(q, d, k)}{w(q, d, k) + w(q, d, k')}$$

We will show in the following how interventional sets can be used to control for unobserved relevance information when estimating propensities.

3.4 Controlling for Unobserved Relevance through Interventional Sets

As we had already seen in Section 3.2, we need to disentangle two unobserved quantities – relevance and examination – when analyzing observed clicks in the PBM. This can be achieved by controlling for relevance through randomization, which was done explicitly in Section 3.2 via $Swap(1, k)$-interventions. The following shows that interventional sets $S_{k,k'}$ provide analogous control for any pair of ranks (k, k') under condition (1).

For each interventional set $S_{k,k'}$, with $k \neq k' \in [M]$, we can now define the quantities $\hat{c}_k^{k,k'}$ (and $\hat{c}_{k'}^{k,k'}$) which can be thought of as

the rate of clicks in position k (and in position k'):

$$\hat{c}_k^{k,k'} := \sum_{i=1}^{m} \sum_{j=1}^{n_i} \sum_{d \in \Omega(q_i^j)} \mathbb{1}_{[(q_i^j, d) \in S_{k,k'}]} \mathbb{1}_{[\mathrm{rk}(d|y_i^j) = k]} \frac{c_i^j(d)}{w(q_i^j, d, k)}.$$

$$\hat{c}_{k'}^{k,k'} := \sum_{i=1}^{m} \sum_{j=1}^{n_i} \sum_{d \in \Omega(q_i^j)} \mathbb{1}_{[(q_i^j, d) \in S_{k,k'}]} \mathbb{1}_{[\mathrm{rk}(d|y_i^j) = k']} \frac{c_i^j(d)}{w(q_i^j, d, k')}.$$

The definition normalizes the observed clicks by dividing with $w(q_i^j, d, k)$, which accounts for non-uniform assignment probabilities. Note that $w(q_i^j, d, k)$ is non-zero whenever the first indicator is true, such that we never divide a non-zero quantity by zero (and we define $0/0 := 0$). Intuitively, $\hat{c}_k^{k,k'}$ and $\hat{c}_{k'}^{k,k'}$ capture the weighted click-through rate at position k and k' restricted to (k, k')-interventional (query, document) pairs, where the weights $w(q, d, k)$ account for the imbalance in applying the intervention of putting document d at position k vs k' for query q.

The following shows that $\hat{c}_k^{k,k'}$ and $\hat{c}_{k'}^{k,k'}$ are proportional to the true clickthrough rate at positions k and k' in expectation, conditioned on the number of relevant documents in the interventional set $S_{k,k'}$.

PROPOSITION 1. *For the PBM model, i.i.d. queries $q \sim \mathrm{Pr}(Q)$, and under the condition in (1), the expectations of $\hat{c}_k^{k,k'}$ and $\hat{c}_{k'}^{k,k'}$ are*

$$\mathbb{E}_{q,c}[\hat{c}_k^{k,k'}] = p_k r_{k,k'}$$
$$\mathbb{E}_{q,c}[\hat{c}_{k'}^{k,k'}] = p_{k'} r_{k,k'}$$

where $r_{k,k'} = \mathbb{E}_q \left[\sum_{d \in \Omega(q)} \mathbb{1}_{[(q,d) \in S_{k,k'}]} \mathrm{rel}(q, d) \right]$.

PROOF. We only detail the proof for $\hat{c}_k^{k,k'}$, since the proof for $\hat{c}_{k'}^{k,k'}$ is analogous.

$$\mathbb{E}_{q,c}[\hat{c}_k^{k,k'}]$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{n_i} \sum_{q \in Q} \mathrm{Pr}(q) \sum_{d \in \Omega(q)} \mathbb{1}_{[(q,d) \in S_{k,k'}]} \mathbb{1}_{[\mathrm{rk}(d|f_i(q)) = k]} \frac{\mathbb{E}_c[c(d)]}{w(q, d, k)}$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{n_i} \sum_{q \in Q} \mathrm{Pr}(q) \sum_{d \in \Omega(q)} \mathbb{1}_{[(q,d) \in S_{k,k'}]} \mathbb{1}_{[\mathrm{rk}(d|f_i(q)) = k]} \frac{p_k \, \mathrm{rel}(q, d)}{w(q, d, k)}$$

$$= p_k \sum_{q \in Q} \mathrm{Pr}(q) \sum_{d \in \Omega(q)} \mathbb{1}_{[(q,d) \in S_{k,k'}]} \mathrm{rel}(q, d) \frac{\sum_{i=1}^{m} \sum_{j=1}^{n_i} \mathbb{1}_{[\mathrm{rk}(d|f_i(q)) = k]}}{w(q, d, k)}$$

$$= p_k \mathbb{E}_q[\sum_{d \in \Omega(q)} \mathbb{1}_{[(q,d) \in S_{k,k'}]} \mathrm{rel}(q, d) \frac{\sum_{i=1}^{m} n_i \mathbb{1}_{[\mathrm{rk}(d|f_i(q)) = k]}}{w(q, d, k)}]$$

$$= p_k \mathbb{E}_q[\sum_{d \in \Omega(q)} \mathbb{1}_{[(q,d) \in S_{k,k'}]} \mathrm{rel}(q, d)]$$

The first equality follows from the i.i.d. assumption of condition (1) and that $y_i^j = f_i(q_i^j)$ by definition, the second from the PBM definition, and the third by taking the inner terms common. □

The quantity $r_{k,k'}$ is related to the average relevance of the documents in the interventional set $S_{k,k'}$ (also see Section 4.2).

While $r_{k,k'}$ is unobserved, it is shared between both $\hat{c}_k^{k,k'}$ and $\hat{c}_{k'}^{k,k'}$. We can thus get the relative propensity between positions k and k' as

$$\frac{p_k}{p_k'} = \frac{p_k r_{k,k'}}{p_{k'} r_{k,k'}} = \frac{\mathbb{E}_{q,c}[\hat{c}_k^{k,k'}]}{\mathbb{E}_{q,c}[\hat{c}_{k'}^{k,k'}]}.$$

We are now in a position to define specific estimators for the relative propensities based on the interventional sets.

4 PROPENSITY ESTIMATORS FOR INTERVENTIONAL SETS

This section defines relative propensity estimators that use interventional sets. The first set of estimators, which we call local estimators, are straightforward adaptations of estimators that had been proposed for data from explicit interventions [18, 29]. However, these local estimators ignore much of the available information when harvesting interventions, and we thus develop a new global estimator that exploits information from all interventional sets $S_{k,k'}$.

4.1 Local Estimators

Since the click counts $\hat{c}_k^{k,k'}$ and $\hat{c}_{k'}^{k,k'}$ can be treated just like data from an explicit intervention, the same estimators apply. In particular, we observe the interventional sets $S_{1,k}$ and can thus use the same estimator that we previously used for the explicit $Swap(1, k)$ experiment [18]. We call this the **PivotOne** estimator

$$\frac{\hat{p}_k}{\hat{p}_1} = \frac{\hat{c}_k^{1,k}}{\hat{c}_1^{1,k}}$$

We can similarly adapt the estimator used in [29], which uses a chain of swaps between adjacent positions in the ranking. We call this the **AdjacentChain** estimator

$$\frac{\hat{p}_k}{\hat{p}_1} = \frac{\hat{c}_2^{1,2}}{\hat{c}_1^{1,2}} \cdot \frac{\hat{c}_3^{2,3}}{\hat{c}_2^{2,3}} \cdot \ldots \cdot \frac{\hat{c}_k^{k-1,k}}{\hat{c}_{k-1}^{k-1,k}}$$

It is easy to see that both estimators are statistically consistent under mild conditions, most importantly that each relevant interventional set has non-zero support such that $\hat{c}_k^{k,k'}$ and $\hat{c}_{k'}^{k,k'}$ concentrate to their expectations.

4.2 Global AllPairs Estimator

A key shortcoming of the local estimators is that they only use a small part of the available information, and that they ignore the data from most interventional sets $S_{k,k'}$. To overcome this shortcoming, we developed a new extremum estimator called **AllPairs** that resembles a maximum likelihood objective over all interventional sets. In order to formulate the training objective of the AllPairs estimator, we first need to define a quantity that is analogous to

$\hat{c}_k^{k,k'}$ and $\hat{c}_{k'}^{k,k'}$, but counting the non-click events:

$$\neg c_k^{k,k'} := \sum_{i=1}^m \sum_{j=1}^{n_i} \sum_{d \in \Omega(q_i^j)} \mathbb{1}_{[(q_i^j,d) \in S_{k,k'}]} \mathbb{1}_{[\mathrm{rk}(d|y_i^j)=k]} \frac{1 - c_i^j(d)}{w(q_i^j, d, k)}$$

$$\neg c_{k'}^{k,k'} := \sum_{i=1}^m \sum_{j=1}^{n_i} \sum_{d \in \Omega(q_i^j)} \mathbb{1}_{[(q_i^j,d) \in S_{k,k'}]} \mathbb{1}_{[\mathrm{rk}(d|y_i^j)=k']} \frac{1 - c_i^j(d)}{w(q_i^j, d, k')}.$$

Analogous to $\hat{c}_k^{k,k'}$ and $\hat{c}_{k'}^{k,k'}$, both quantities have the desired expectation.

PROPOSITION 2. *For the PBM model, i.i.d. queries $q \sim \Pr(Q)$, and under the condition in* (1), *the expectations of $\neg \hat{c}_k^{k,k'}$ and $\neg \hat{c}_{k'}^{k,k'}$ are*

$$\mathbb{E}_{q,c}[\neg \hat{c}_k^{k,k'}] = N_{k,k'} - p_k r_{k,k'}$$

$$\mathbb{E}_{q,c}[\neg \hat{c}_{k'}^{k,k'}] = N_{k,k'} - p_k r_{k,k'}$$

where $r_{k,k'} = \mathbb{E}_q \left[\sum_{d \in \Omega(q)} \mathbb{1}_{[(q,d) \in S_{k,k'}]} \mathrm{rel}(q,d) \right]$ and $N_{k,k'} = \mathbb{E}_q[\sum_{d \in \Omega(q)} \mathbb{1}_{[(q,d) \in S_{k,k'}]}]$.

PROOF. The proof is analogous to Proposition 1 and thus omitted. □

The way we constructed the interventional sets, the expected relevances $r_{k,k'}$ are not necessarily normalized to be within the interval $[0, 1]$. It is therefore more convenient to instead consider the normalized version of $r_{k,k'}$

$$\bar{r}_{k,k'} \equiv \frac{r_{k,k'}}{N_{k,k'}} \in [0, 1],$$

using the definitions from Propositions 1 and 2. Under this normalization, we have that $\mathbb{E}[\hat{c}_k^{k,k'}] = p_k \bar{r}_{k,k'} N_{k,k'}$ and $\mathbb{E}[\neg c_k^{k,k'}] = (1 - p_k \bar{r}_{k,k'}) N_{k,k'}$ and similarly for $\hat{c}_k'^{k,k'}$ and $\neg c_k^{k,k'}$. We model this normalized $\bar{r}_{k,k'}$ in the AllPairs estimator.

We can now formulate the training objective of the AllPairs estimator. The following objective needs to be maximized with respect to $\hat{p}_k \in [0, 1]$ and $\hat{r}_{k,k'} = \hat{r}_{k',k} \in [0, 1]$ (since $\bar{r}_{k,k'} = \bar{r}_{k',k}$ by definition)

$$(\hat{p}, \hat{r}) = \underset{p, r}{\mathrm{argmax}} \sum_{k \neq k' \in [M]} \hat{c}_k^{k,k'} \log(\hat{p}_k \hat{r}_{k,k'}) + \neg \hat{c}_k^{k,k'} \log(1 - \hat{p}_k \hat{r}_{k,k'}).$$

This optimization problem can be interpreted as Weighted Cross-Entropy Maximization for estimating the distribution $p_k \bar{r}_{k,k'}$ using weighted samples $\hat{c}_k^{k,k'}$ and $\neg \hat{c}_k^{k,k'}$. The weighting by $N_{k,k'}$ ensures that the contribution of each aggregated click-through sample is proportional to the size of its interventional set.

From the solution of this optimization problem, $(\hat{p}, \hat{r})$, we only need $\hat{p}$ while the matrix $\hat{r}$ can be discarded. To get normalized propensities relative to rank 1, we compute $\hat{p}_k/\hat{p}_1$. Note that the optimization problem is quite small, as it uses only $O(M^2)$ variables and $2 * M * (M - 1)$ terms in the objective.

Unlike the local estimators, the AllPairs approach integrates all data from every interventional set. We will evaluate empirically how much statistical efficiency is gained by taking this global approach compared to the local estimators.

5 EMPIRICAL EVALUATION

We take a two-pronged approach to evaluating our intervention-harvesting technique and the estimators. First, we fielded them on two real-world systems – the Arxiv Full-Text Search Engine and the Google Drive Search – to gain insight into their practical effectiveness. Second, we augment these real-world experiments with simulation studies using synthetically generated click data, where we can explore the behavior of our method over the whole spectrum of settings (e.g. varying ranker similarity, presentation-bias severity, and click noise).

5.1 Real-World Evaluation: Arxiv Full-Text Search

We conducted a controlled experiment on the Arxiv Full-Text Search Engine[1], where we compare the results from the AllPairs estimator using Intervention Harvesting with the results from a gold-standard intervention experiment as described below. We used three ranking functions $\{f_1, f_2, f_3\}$ for defining the interventional sets, which were generated by using different learning methods and datasets.

Gold Standard. Since we do not know the true propensities for the Arxiv Full-Text Search Engine, we use the $Swap(1, k)$ interventions and estimator described in Section 3.2 as our gold standard. With probability 0.5, an incoming query is assigned to generating data towards a swap experiment. For each assigned query, a ranking function f_i is chosen uniformly at random, and its rank 1 is swapped uniformly with rank $k \in \{1, .., 21\}$ before it is presented to the user. The $\hat{c}_1^{1,k}$ and $\hat{c}_k^{1,k}$ are computed over all three f_i.

Intervention Harvesting. For the other half of the incoming queries, we uniformly pick a ranking function f_i and present its results without further intervention. From the data in these conditions, we then compute the interventional sets $S_{k,k'}$ for $k \neq k' \in \{1, .., 21\}$ and their resulting $\hat{c}_k^{k,k'}$, $\hat{c}_{k'}^{k,k'}$, $\neg\hat{c}_k^{k,k'}$ and $\neg\hat{c}_{k'}^{k,k'}$. These are then used in the AllPairs estimator from Section 4.2.

5.1.1 Results. Data for all six conditions was collected simultaneously between May 14, 2018 and August 1, 2018 to avoid confounding due to shift in the query distribution for maximum validity of the experiment. About 53,000 queries and 25,600 clicks were collected, about half for the Swap Experiment and half for the Interventional Set method.

The estimated propensity curves for the gold-standard swap experiment and for the AllPairs estimator are shown in Figure 1. The shaded region for each curve corresponds to a 95% confidence interval calculated from 1000 bootstrap samples. As we can see, the curves follow a similar trend for each position and AllPairs mostly lies within the confidence interval of the gold-standard curve. However, the confidence interval for the AllPairs method is substantially tighter than for the swap experiment, indicating that AllPairs is not only less intrusive (no swap interventions) but also statistically more efficient given the same number of queries.

The following provide further insight into this gain in efficiency. While we conducted interleaving experiments [5] to confirm that the three ranking functions $\{f_1, f_2, f_3\}$ provide similar ranking accuracy, we found that the three rankers tend to assign documents

Table 1: Size of the interventional sets $S_{k,k'}$ for Arxiv (showing only top 10 positions).

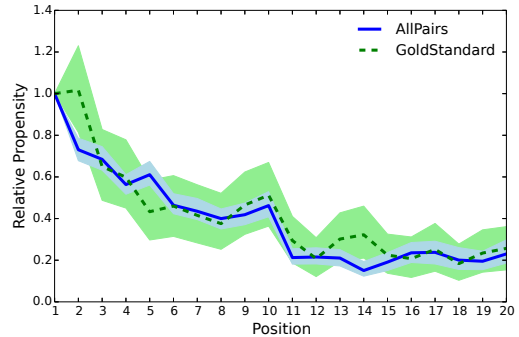| rank k									rank k'
1	2	3	4	5	6	7	8	9	
13625	8340	6723	5231	3966	3051	2656	2274	2015	2
-	9039	7692	5994	5053	3588	2675	2861	2244	3
-	-	8555	6994	5573	4117	3368	3321	2361	4
-	-	-	6783	5345	5040	3849	3427	3614	5
-	-	-	-	6290	4809	4058	4126	3489	6
-	-	-	-	-	5466	4746	3935	3294	7
-	-	-	-	-	-	5425	4092	3692	8
-	-	-	-	-	-	-	4258	3930	9
-	-	-	-	-	-	-	-	3719	10

Figure 1: Estimated propensity curves for Arxiv. Shaded regions correspond to a 95% confidence interval.

to different ranks. In fact, only 7.39% of the documents were ranked at the same rank by all three rankers, such that about 93% of the documents contributed to an interventional set and thus became meaningful training data for AllPairs. Table 1 shows the size of the interventional sets $S_{k,k'}$ for the top 10 positions. While local swaps are most common, the rankers frequently rank the same document at substantially different ranks.

5.2 Real-World Evaluation: Google Drive Search

We conducted a second real-world experiment on the search for Google Drive. The service uses an overlay to show results as users type. The overlay disappears when a click on the overlay happens. Thus, each query has at most a single click. The overlay displays at most 5 results for each query and all the displayed results are logged with their position information. Again, we compare the results from the AllPairs estimator against a gold-standard intervention experiment.

Gold Standard. We follow [29] and use $Swap(k, k + 1)$ interventions along with the AdjacentChain estimator as our gold standard.

Intervention Harvesting. In an A/B test, users were randomly assigned either to the production ranker f_1 or a new ranker f_2. Based on these two rankers the interventional sets $S_{k,k'}$ with $k \neq k' \in \{1, .., 5\}$ were computed and provided to the AllPairs estimator.
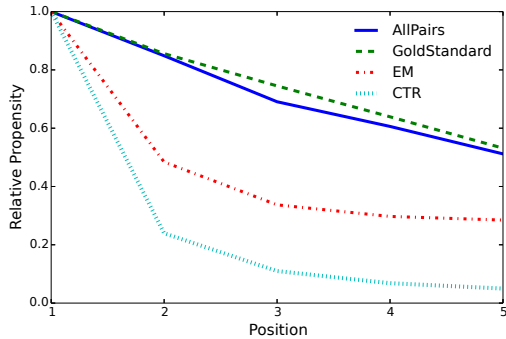
Figure 2: Estimated propensity curves for Google Drive.

Table 2: Size of the interventional sets $S_{k,k'}$ for Google Drive.

rank k				rank k'
1	2	3	4	
19,516	2,203	579	244	2
-	48,576	10,630	3,780	3
-	-	73,170	21,037	4
-	-	-	92,343	5

Results. During two weeks in April 2018, a total of 877,689 queries were collected. Note that data for the gold-standard swap experiment was collected at a different time and for a different ranking function, but is is reasonable to assume that the propensity curve has not changed. The estimated propensity curves for the AllPairs estimator and the gold-standard swap experiment are shown in Figure 2. The AllPairs estimates closely resemble the gold standard as desired. For comparison, we also computed propensity estimates via the regression EM from [29] on the same data set as AllPairs. Those estimates are substantially off from the gold standard. Even further off is the naive method of using the empirical clickthrough rate (CTR) as an estimate for p_k without any experimental control for relevance. Overall, we find that Intervention Harvesting with AllPairs provides superior results that are close to the gold standard.

In some ways, the Arxiv experiment and the Google Drive experiment covered two substantially different use cases. For Arxiv, the rankings were substantially different, while for Google Drive the two ranking functions were typically quite close. In fact, the two rankers for Google Drive provided identical top-5 rankings for 75.26% of the queries, which thus did not contribute any useful data to the interventional sets. And even among the queries that lead to different rankings, 74.5% of the results were at the same rank in both ranking functions. This means much less interventional-set data was generated per query, and Table 2 further shows that most of the swaps were quite local. It is therefore reassuring that AllPairs nevertheless provides accurate estimates, and that it works well for both Arxiv and Google Drive despite these differences.

5.3 Robustness Analysis: Yahoo LTR Challenge

While the two real-world experiments provide validation for the applicability of the method, these are just two data-points in a large spectrum of possible settings. We therefore now evaluate
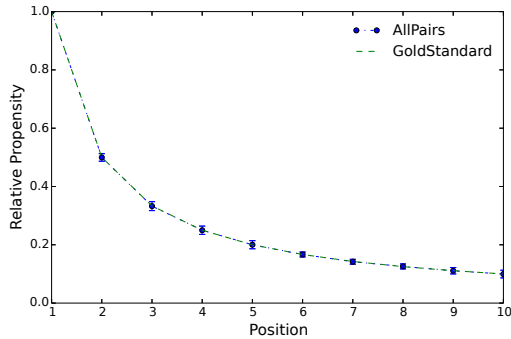

Figure 3: Estimated propensity curve for synthetic click data derived from the Yahoo LTR dataset ($\eta = 1$, $\epsilon_- = 0.1$, $n_i = 99720$, frac = 0.02, overlap = 0.8).

the robustness of the method on synthetic click data, where we know the true propensity curve by construction and can control the properties of the data with respect to all relevant parameters (e.g. noise, ranker similarity, data-set size).

Experiment Setup. We generated synthetic click data according to the following methodology that closely matches that in [18]. In particular, we use the Yahoo LTR Challenge data, which comes with manual relevance judgments. Using these relevance judgments, clicks were generated by simulating the Position-Based Model with propensities that decay with the presented rank via $p_r = \left(\frac{1}{r}\right)^{\eta}$. The parameter η controls the severity of bias, with higher values causing greater position bias. We also introduced noise into the clicks by allowing some irrelevant documents to be clicked. Specifically, an irrelevant document ranked at position r by the production ranker is clicked with probability p_r times ϵ_- whereas a relevant document is clicked with probability p_r. For simplicity (and without loss of generality), we used click logs from two rankers in each experiment setting. Rankers were obtained by training Ranking SVMs on random samples of queries with their manual relevance judgments. The "similarity" of the two rankers was controlled by varying the degree of overlap in their respective training sets. We evaluate the estimation accuracy via the Mean Squared Error (MSE) of the estimated inverse relative propensity weights, $MSE = \frac{\sum_{i=1}^{M}(\hat{p}_1/\hat{p}_i - p_1/p_i)^2}{M}$, since these inverse propensity weights better reflect how inaccurate propensity estimates impact the IPS estimator [18]. We estimate propensities up to rank $M = 10$. Error bars indicate the standard deviation over 6 independent runs (except in Figure 3 as described below). If not mentioned otherwise, the number of simulated queries per ranker is $n_i = 99720$ (obtained by 5 sweeps of the 19944 queries in the Yahoo LTR training set), and we use $\eta = 1$ and $\epsilon_- = 0.1$.

5.3.1 Does AllPairs recover the true propensity curve? Figure 3 shows the estimated propensities of AllPairs in comparison to the true propensity curve of ($\eta = 1$) that is known by construction. All parameters of the simulation experiment are kept at the defaults as stated above. As expected, AllPairs perfectly recovers the true propensities. The error bars show the 99% confidence intervals that
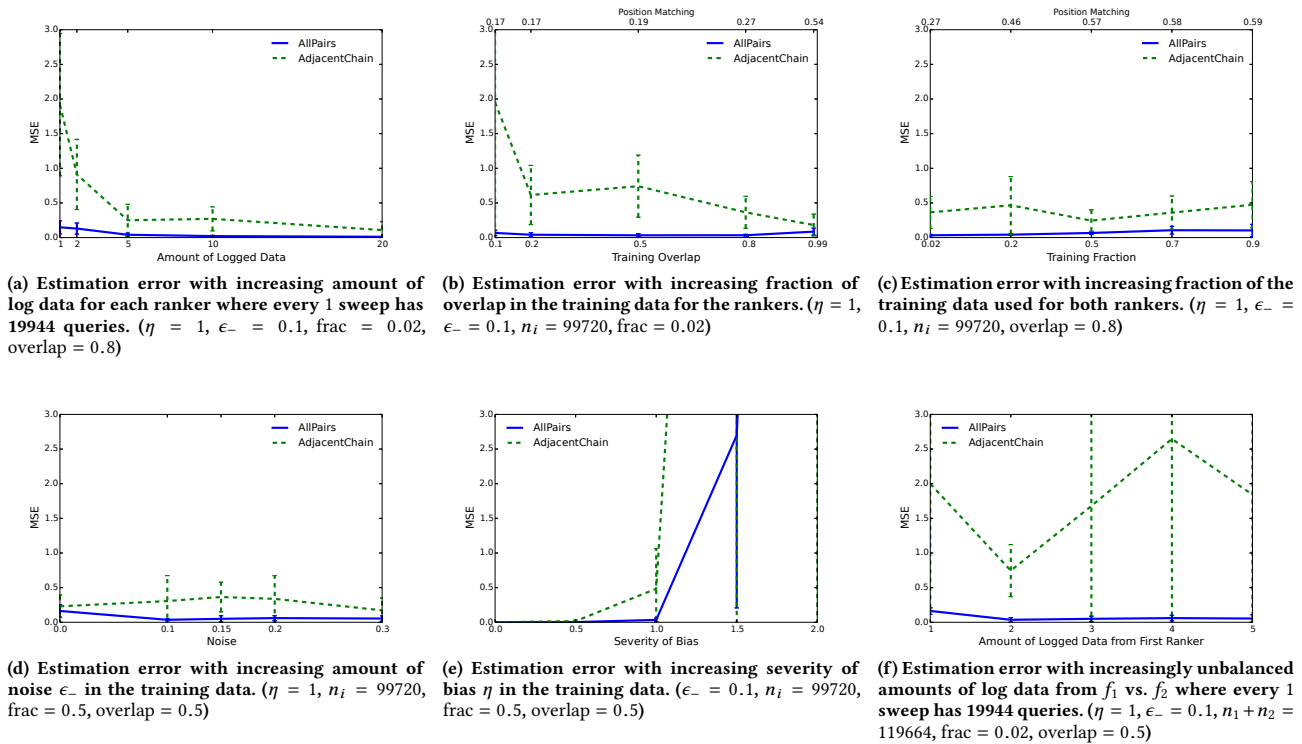
(a) Estimation error with increasing amount of log data for each ranker where every 1 sweep has 19944 queries. ($\eta = 1$, $\epsilon_- = 0.1$, frac = 0.02, overlap = 0.8)

(b) Estimation error with increasing fraction of overlap in the training data for the rankers. ($\eta = 1$, $\epsilon_- = 0.1$, $n_i = 99720$, frac = 0.02)

(c) Estimation error with increasing fraction of the training data used for both rankers. ($\eta = 1$, $\epsilon_- = 0.1$, $n_i = 99720$, overlap = 0.8)

(d) Estimation error with increasing amount of noise ϵ_- in the training data. ($\eta = 1$, $n_i = 99720$, frac = 0.5, overlap = 0.5)

(e) Estimation error with increasing severity of bias η in the training data. ($\epsilon_- = 0.1$, $n_i = 99720$, frac = 0.5, overlap = 0.5)

(f) Estimation error with increasingly unbalanced amounts of log data from f_1 vs. f_2 where every 1 sweep has 19944 queries. ($\eta = 1$, $\epsilon_- = 0.1$, $n_1 + n_2 = 119664$, frac = 0.02, overlap = 0.5)

Figure 4: Robustness experiments on the synthetic click data derived from the Yahoo LTR dataset.

are computed via the standard deviation estimated from re-running the simulation experiment 20 times.

5.3.2   How much data is needed? While the previous section showed that Intervention Harvesting with the AllPairs estimator converges to the true propensities, we now analyze the speed of convergence. Figure 4 (a) shows that AllPairs provides good estimates even for modest amounts of data. As a baseline for comparison, we also show the MSE of the AdjacentChain estimator, which is substantially worse and requires at least one order of magnitude more data to achieve the same MSE as AllPairs. We also explored the use of the PivotOne estimator, but do not report its results since they are typically worse than those of AdjacentChain.

5.3.3   How different should the ranking functions be? If the ranking functions f_i are all identical, then Intervention Harvesting cannot produce any data. So, how different do the rankers f_i need to be? To vary ranker similarity, we trained pairs of rankers with increasing overlap (from 1% to 99%) in their training sets. In Figure 4 (b), we see that the estimation accuracy remains quite robust even as the rankers become increasingly similar due to the overlap in the data they are trained on. The top of the plot shows the similarity in terms of the fraction of documents at the same rank in both rankers averaged across all the queries. As expected, the error goes up when the rankers are very similar since then they tend to put documents at the same position, leading to fewer interventional pairs. Interestingly, the error is also relatively higher when the rankers are too dissimilar. This is because when the candidate sets are larger

than 10, the dissimilarity in the rankers causes many interventions to be discarded since they often go beyond rank 10. Note that AdjacentChain benefits from ranker similarity, since it focuses the interventional set data on $S_{k,k+1}$. However, AdjacentChain at best matches the performance of AllPairs, but never outperforms it.

5.3.4   How important is the quality of the rankers? Another way of controlling ranker similarity is to increase the total number of training examples for both. Figure 4 (c) shows the result of this experiment. Again, AllPairs shows robust performance over the whole spectrum of settings and substantially improves over AdjacentChain.

5.3.5   How does click noise impact estimation accuracy? Clicks in any real-world setting will be noisy, and we thus want to explore the robustness of AllPairs with respect to noise. Note that noise (which can be seen simply as an alternative vector of relevances $rel(q, d)$ in our model) should have no influence on the propensity estimates via Information Harvesting, as long as the noise is not confounded by rank (which is guaranteed in the PBM). Figure 4 (d) verifies that Information Harvesting estimates are indeed stable over different levels of ϵ_- noise.

5.3.6   How does bias severity impact estimation accuracy? Figure 4 (e) explores the behavior of the estimators when we vary the steepness of the propensity curve via η. The AllPairs method performs better than AdjacentChain, but the MSE of both methods increases sharply when the propensity curve gets steep. The explanation for

this is twofold. First, the increase is partly an artifact of the MSE error measure on the inverse propensities. Steep curves lead to small propensities, which in turns generates large inverse propensities that provide opportunity for large MSE. Second, for steep propensity curves the bottom ranks receive only few clicks such that there is not enough data for estimating their propensities reliably.

5.3.7 How robust is the method to imbalanced datasets? Finally, we explore the situation where we may have more log data from one ranker than from the other. Figure 4 (f) shows MSE when we vary the amount of log data for f_1 while keeping the total amount of data constant at 119, 664 queries (i.e. 6 sweeps). The plot shows that AllPairs is robust to such imbalances.

6 CONCLUSION

We presented the idea of Intervention Harvesting, allowing the use of multiple historic loggers for generating interventional data under mild assumptions. We showed how this idea can be used to control for relevance, providing the first method for propensity estimation in the PBM that does not require intrusive interventions, a relevance model, or repeat queries. In particular, we propose the AllPairs estimator for combining all intervention data, which we find to provide superior propensity estimation accuracy compared to existing local estimators over a wide spectrum of settings. Beyond these contributions, the paper opens an interesting space of research questions for how other intervention data can be harvested and what other estimation problems can be addressed in this way.

7 ACKNOWLEDGMENTS

REFERENCES

[1] Aman Agarwal, Soumya Basu, Tobias Schnabel, and Thorsten Joachims. 2017. Effective Evaluation using Logged Bandit Feedback from Multiple Loggers. In *Proc. of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 687–696.

[2] Aman Agarwal, Ivan Zaitsev, and Thorsten Joachims. 2018. Counterfactual Learning-to-Rank for Additive Metrics and Deep Models. In *ICML Workshop on Machine Learning for Causal Inference, Counterfactual Prediction, and Autonomous Action (CausalML)*.

[3] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased Learning to Rank with Unbiased Propensity Estimation. In *Proc. of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*. 385–394.

[4] Ben Carterette and Praveen Chandar. 2018. Offline Comparative Evaluation with Incremental, Minimally-Invasive Online Feedback. In *Proc. of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*. 705–714.

[5] Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. 2012. Large-Scale Validation and Analysis of Interleaved Search Evaluation. *ACM Transactions on Information Systems (TOIS)* 30, 1 (2012), 6:1–6:41.

[6] Olivier Chapelle and Ya Zhang. 2009. A dynamic bayesian network click model for web search ranking. In *Proc. of the 18th International Conference on World Wide Web (WWW)*. 1–10.

[7] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. *Click Models for Web Search*. Morgan & Claypool Publishers.

[8] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An Experimental Comparison of Click Position-bias Models. In *Proc. of the 1st International Conference on Web Search and Web Data Mining (WSDM)*. 87–94.

[9] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)* 39 (1977), 1–38. Issue 1.

[10] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly Robust Policy Evaluation and Learning. In *Proc. of the 28th International Conference on Machine Learning (ICML)*. 1097–1104.

[11] Georges E. Dupret and Benjamin Piwowarski. 2008. A User Browsing Model to Predict Search Engine Click Data from Past Observations. In *Proc. of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 331–338.

[12] Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Wang, and Christos Faloutsos. 2009. Click Chain Model in Web Search. In *Proc. of the 18th International Conference on World Wide Web (WWW)*. 11–20.

[13] G. Imbens and D. Rubin. 2015. *Causal Inference for Statistics, Social, and Biomedical Sciences*. Cambridge University Press.

[14] Thorsten Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. In *Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 133–142.

[15] Thorsten Joachims, Laura A. Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately Interpreting Clickthrough Data as Implicit Feedback. In *Proc. of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 154–161.

[16] Thorsten Joachims, Laura A. Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. 2007. Evaluating the Accuracy of Implicit Feedback from Clicks and Query Reformulations in Web Search. *ACM Transactions on Information Systems (TOIS)* 25, 2, Article 7 (April 2007).

[17] Thorsten Joachims, Adith Swaminathan, and Maarten de Rijke. 2018. Deep Learning with Logged Bandit Feedback. In *6th International Conference on Learning Representations (ICLR)*.

[18] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Proc. of the 10th ACM International Conference on Web Search and Data Mining, (WSDM)*. 781–789.

[19] John Langford, Alexander Strehl, and Jennifer Wortman. 2008. Exploration Scavenging. In *Proc. of the 25th International Conference on Machine Learning (ICML)*. 528–535.

[20] Lihong Li, Shunbao Chen, Jim Kleban, and Ankur Gupta. 2014. Counterfactual Estimation and Optimization of Click Metrics for Search Engines. *CoRR* abs/1403.1891 (2014).

[21] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. 2011. Unbiased Offline Evaluation of Contextual-bandit-based News Article Recommendation Algorithms. In *Proc. of the 4th International Conference on Web Search and Web Data Mining (WSDM)*. 297–306.

[22] Maeve O'Brien and Mark T Keane. 2006. Modeling result–list searching in the World Wide Web: The role of relevance topologies and trust bias. In *Proc. of the 28th Annual Conference of the Cognitive Science Society (CogSci)*. 1–881.

[23] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting Clicks: Estimating the Click-through Rate for New Ads. In *Proc. of the 16th International Conference on World Wide Web (WWW)*. 521–530.

[24] Paul R. Rosenbaum and Donald B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70, 1 (1983), 41–55.

[25] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as Treatments: Debiasing Learning and Evaluation. In *Proc. of the 33rd International Conference on Machine Learning (ICML)*. 1670–1679.

[26] Adith Swaminathan and Thorsten Joachims. 2015. Batch Learning from Logged Bandit Feedback through Counterfactual Risk Minimization. *Journal of Machine Learning Research (JMLR)* 16 (Sep 2015), 1731–1755.

[27] Adith Swaminathan and Thorsten Joachims. 2015. The Self-Normalized Estimator for Counterfactual Learning. In *Proc. of the 28th International Conference on Neural Information Processing Systems (NIPS)*. 3231–3239.

[28] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to Rank with Selection Bias in Personal Search. In *Proc. of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 115–124.

[29] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position Bias Estimation for Unbiased Learning to Rank in Personal Search. In *Proc. of the 11th ACM International Conference on Web Search and Data Mining (WSDM)*. 610–618.

[30] Yisong Yue, Rajan Patel, and Hein Roehrig. 2010. Beyond position bias: examining result attractiveness as a source of presentation bias in clickthrough data. In *Proc. of the 19th International Conference on World Wide Web (WWW)*. 1011–1018.