Non-Clicks Mean Irrelevant? Propensity Ratio Scoring As a Correction

Nan Wang[1], Zhen Qin[2], Xuanhui Wang[2], Hongning Wang[1]
[1]University of Virginia, Charlottesville, VA
[2]Google Research, Mountain View, CA
nw6a@virginia.edu,{zhenqin,xuanhui}@google.com,hw5x@virginia.edu

ABSTRACT

Recent advances in unbiased learning to rank (LTR) count on Inverse Propensity Scoring (IPS) to eliminate bias in implicit feedback. Though theoretically sound in correcting the bias introduced by treating clicked documents as relevant, IPS ignores the bias caused by (implicitly) treating non-clicked ones as irrelevant. In this work, we first rigorously prove that such use of click data leads to unnecessary pairwise comparisons between relevant documents, which prevent unbiased ranker optimization. Based on the proof, we derive a simple yet well justified new weighting scheme, called Propensity Ratio Scoring (PRS), which provides treatments on both clicks and non-clicks. Besides correcting the bias in clicks, PRS avoids relevant-relevant document comparisons in LTR training and enjoys a lower variability. Our extensive empirical evaluations confirm that PRS ensures a more effective use of click data and improved performance in both synthetic data from a set of LTR benchmarks, as well as in the real-world large-scale data from GMail search.

KEYWORDS

Unbiased learning to rank, implicit feedback

ACM Reference Format:
Nan Wang, Zhen Qin, Xuanhui Wang, Hongning Wang. 2021. Non-Clicks Mean Irrelevant? Propensity Ratio Scoring As a Correction. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3437963.3441798

1 INTRODUCTION

Implicit feedback from users, such as clicks, provides an abundant resource of relevance signals for learning to rank (LTR) [16]. But such data is also notoriously biased for various reasons, e.g., the most notable position bias [17], which is known to distort LTR training if not properly handled [17, 18, 33].

Inverse Propensity Scoring (IPS) [19] has emerged as a mainstream solution to debias implicit feedback for LTR. It provides an unbiased estimate of the ranking metric of interest, such as Average Relevance Position (ARP, termed as *rank* in [19]), by reweighing the clicked documents. However, the unbiasedness of IPS is only maintained in the estimation of the ranking metrics, rather than in the actual ranker optimization under such metrics. A serious gap is introduced when one evaluates those ranking metrics using click data for ranker optimization. Specifically, due to the non-continuous nature of most ranking metrics, the optimization has to be performed on induced loss of those metrics in practice. As shown in [31], most popular ranking metrics, such as ARP and NDCG, can be decomposed into pairwise comparisons. Thus pairwise loss is introduced for continuous approximation and ranker optimization under those metrics. For example, in a ranked list, each clicked document is compared against all others to compute the hinge loss for ARP as in Propensity SVM-Rank [19], or against only non-clicked ones to compute the lambda loss for NDCG as in Unbiased LambdaMART [13]. The mapping from a ranking metric estimator to its induced loss by clicks opens the door to the detrimental deficiency of IPS-based unbiased LTR, which is serious but largely ignored.

	Observed	Unobserved
Relevant	**Clicked** (R&O)	**Non-clicked** (R&U)
Irrelevant	**Non-clicked** (I&O)	**Non-clicked** (I&U)

Figure 1: Decomposition of click data. We use 'R', 'I' to denote relevant or irrelevant; and 'O', 'U' for observed or unobserved. Only documents in the R&O part are clicked.

In this work, we rigorously prove the gap between the IPS estimators of the ranking metrics and the practical ranker optimization on the induced losses, in terms of unbiasedness, for the first time. In particular, we show that comparisons between pairs of the same relevance label only contribute a constant term to the evaluation of ranking metrics, regardless of their ranked positions or ranking scores. Thus one should avoid counting loss on such pairs. Because IPS [1, 3, 19] can only correct bias in using clicked documents to measure relevance, but non-click does not necessarily stand for irrelevance (it can be relevant but not observed), existing IPS-based solutions inevitably count loss on relevant-relevant document comparisons. This introduces an irreducible gap between the ranking metric one expects to optimize and the actual loss one ends up with.

To illustrate the problem in a more intuitive way, we decompose click data as shown in Figure 1. In a noise-free setting, click happens if and only if a relevant document is observed (i.e., R&O), while all others are recorded as non-clicks (i.e., R&U, I&O and I&U in the figure). This leads to an *asymmetric* relation between clicked and non-clicked documents. When using click data for LTR training, IPS can solely correct bias in the clicked part, so that the total loss can be extended from clicked documents to all relevant documents

WSDM '21, March 8–12, 2021, Virtual Event, Israel

ACM ISBN 978-1-4503-8297-7/21/03.
https://doi.org/10.1145/3437963.3441798

(R&O+R&U) in terms of expectation. Meanwhile, relevant and unobserved documents (R&U) are also in the non-clicked documents, and are unavoidably used as negative examples in pairwise comparisons in the induced loss for LTR [13, 19]. It leaves the learnt ranker still biased for relevant but non-clicked documents. This deficiency is already reflected in recent empirical studies [15]: only when there is little bias, can the utility of IPS be observed. Because strong bias conceals lots of relevant documents in the non-clicked part, an increased gap is introduced to the loss function and thus distorts the learnt ranker more seriously. No existing work realizes this deficiency of IPS in unbiased LTR yet. A recent work named Unbiased LambdaMART reweighs non-clicked documents with IPS for debiasing purpose [13]. However, as it does not analyze the exact bias in non-clicked documents and still restricts itself in the IPS framework, a rigid assumption has to be imposed that the probability of non-click is proportional to the probability of being irrelevant. This unfortunately is against most click modeling assumptions.

To eliminate the gap, we propose a simple yet effective solution called Propensity Ratio Scoring (PRS) for unbiased LTR with pairwise comparisons. We devise statistical treatments on both clicked and non-clicked documents when forming pairwise loss for metric optimization. Both theoretical and empirical justifications of PRS are provided step by step. More importantly, we show that PRS ensures a more effective use of click data and reduced variability compared to IPS, without any additional requirement on the infrastructure or data collection. We conduct extensive empirical studies based on a set of LTR benchmark datasets to demonstrate the correctness and advantage of PRS. To show its utility in real-world industrial setting, we also evaluate PRS on the large-scale GMail search data, which further confirms its practical significance.

2 RELATED WORK

Click data is a vital resource for LTR model training in modern retrieval systems. But the intrinsic bias, especially the position bias, greatly limits its effective use [18, 33]. Numerous click models [8–11, 28] have been proposed to model the bias in users' click behaviors, so as to extract true relevance labels. But they require repeated observations for reliable relevance inference. Researchers also adopt randomization techniques to eliminate bias [7, 26, 32] when collecting clicks. Though assumption free, randomization degrades the ranking quality during data collection and inevitably hurts user experience.

Recent efforts focus on unbiased LTR directly from biased click data. The key idea is to obtain an unbiased estimator of a ranking metric of interest by leveraging statistical patterns embedded in the click data. Inverse propensity scoring (IPS) [1, 19] is a mainstream solution, which reweighs observational clicks for debiasing. However, IPS is solely applicable to debias clicked documents, and its utility largely depend on whether most relevant documents can be revealed by the clicks. When there is increasing bias or noise, more relevant documents are buried in non-clicked documents, which IPS cannot handle. To make things worse, IPS-based methods treat non-clicked but relevant documents as negative samples for ranker training [13]. This directly leads to its poor performance in practice [15], and we will theoretically prove the cause of this deficiency.

There exist various realizations of the IPS framework for unbiased LTR [1, 3, 13]. Though applied on different model structures, they suffer from the same issues of IPS. Among them, Hu el al. [13] applied IPS to both clicked and non-clicked documents for debiasing. However, this solution ignores the fact that the bias in clicked and non-clicked documents is *asymmetric*, and simply assumes that non-click probability is proportional to the irrelevance probability to apply IPS. In contrast, we carefully analyze the bias introduced in non-clicked documents and derive a justified solution that accounts for the use of both clicks and non-clicks.

3 DIAGNOSIS OF IPS IN PRACTICAL USE

In this section, we first present the general theory of IPS for unbiased ranking metric estimation. Then based on an in-depth discussion of IPS in ranker optimization, we elaborately investigate and disclose the deficiency of IPS in solving the problem.

3.1 IPS for Unbiased Metrics Estimation

Without loss of generality, we are given a set of i.i.d. queries $\boldsymbol{Q}$, where each query q is associated with a list of candidate ranking documents $\{x_i\}_{i=1}^{|q|}$. The goal of LTR is to optimize a ranker π on $\boldsymbol{Q}$ under ranking metrics of interest (e.g., ARP, NDCG, etc). Formally, the empirical risk of π incurred on $\boldsymbol{Q}$ can be obtained as,

$$R(\pi|Q) = \sum_{q\in\boldsymbol{Q}} R(\pi|q) = \sum_{q\in\boldsymbol{Q}} \sum_{x_i: r_q(x_i)=1} \Delta(x_i|\pi_q) \tag{1}$$

where $r_q(x_i)$ is the ground-truth relevance label of document x_i to query q (assuming binary relevance for simplicity), π_q denotes the ranking of documents under query q, and $\Delta(x_i|\pi_q)$ measures the contribution from a relevant document x_i to the ranking metric, e.g., the *rank* of x_i in ARP or the discounted gain of x_i in NDCG.

Unlike the full-information setting where the relevance labels of all documents in π_q are known, r_q is only partially observed in the implicit click feedback, due to various click biases [23]. Simply using clicked documents to realize Eq (1) leads to a biased estimate of the ranking metric.

IPS addresses the bias issue via weighing each clicked document in Eq (1) by the inverse of its observation propensity in the logged ranking $\tilde{\pi}_q$ [19]. To be more specific, denote o_q as a binary vector indicating whether the documents' relevance labels in r_q are observed in $\tilde{\pi}_q$. For each element of o_q, the marginal probability $P(o_q(x_i) = 1|\tilde{\pi}_q)$ is referred to as the observation propensity of document x_i. Consider the deterministic noise-free setting in [19], a document is clicked if and only if it is examined (thus observed) and relevant, i.e., $c_q(x_i) \Leftrightarrow o_q(x_i) \wedge r_q(x_i)$. We can then get an unbiased estimate of $R(\pi_q|q)$ for any new ranking π_q via IPS [14, 24],

$$\begin{aligned} R_{IPS}(\pi_q|q, \tilde{\pi}_q, o_q) &= \sum_{x_i: c_q(x_i)=1} \frac{\Delta(x_i|\pi_q)}{P(o_q(x_i) = 1|\tilde{\pi}_q)} \\ &= \sum_{x_i: o_q(x_i)=1 \wedge r_q(x_i)=1} \frac{\Delta(x_i|\pi_q)}{P(o_q(x_i) = 1|\tilde{\pi}_q)} \end{aligned} \tag{2}$$

where we introduce another binary vector c_q to denote whether a document is clicked under q in $\tilde{\pi}_q$.

Although Eq (2) has been proved to be an unbiased estimate of $R(\pi_q|q)$ for any new ranking π_q: $E_{o_q}\left[R_{IPS}(\pi_q|q, \tilde{\pi}_q, o_q)\right] = R(\pi_q|q)$ [19], a direct optimization of Eq (2) is intractable, due to the non-continuous nature of most ranking metrics. Approximations are thus necessary for optimization [1], which however introduce a gap

from the estimated metric to the induced loss. Next we rigorously examine and analyze the consequence caused by this gap.

3.2 Issues of IPS in Practical Ranker Optimization

In Eq (2), it is important to realize that the individual contribution $\Delta(x_i|\pi_q)$ of a relevant document x_i has to be obtained from its comparisons to other documents in the ranking list π_q, as the ranking position of x_i in r_q depends on how many other documents are predicted to be more relevant than it by the ranker. As shown in [31], this is achieved via pairwise comparisons for a set of popular ranking metrics in practice. For instance, to get the *rank* of a relevant document x_i in π_q, we need to compare its predicted relevance $\hat{r}_q(x_i)$ to all other documents' $\hat{r}_q(x_j)$. One key property in such a pairwise formulation is that only comparisons between two documents of *different* ground-truth relevance labels have influence on the ranking metric evaluation and can thus contribute to the optimization. The comparisons between two documents of the *same* relevance label will only form a constant term, agnostic to their positions or predicted relevance [31].

The gap thus emerges when using continuous approximations on the pairwise comparisons, such as logistic loss [4], hinge loss [16] or LambdaRank loss [6]: the induced loss on comparisons between same-labeled documents become an irreducible term in the total loss that distorts the optimization and leads to sub-optimal results. To theoretically and more explicitly illustrate the issue, we revisit the optimization of Average Relevance Position (ARP), which is analyzed in the first IPS-based LTR solution [19] (but termed as *rank*). The same analysis can be easily applied to the optimization of other ranking metrics, such as NDCG [31].

Assume there are n documents in π_q and the rank of documents starts from 0. Use r_q^i and $\hat{r}_q^i$ to replace $r_q(x_i)$ and $\hat{r}_q(x_i)$ for simplicity. We have the APR evaluated on π_q as:

$$\begin{aligned} ARP =& \frac{1}{n}\sum_{i=1}^{n} rank(x_i|\pi_q)\cdot r_q^i = \frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(r_q^i \mathbb{I}_{\hat{r}_q^i<\hat{r}_q^j} + r_q^j \mathbb{I}_{\hat{r}_q^j<\hat{r}_q^i}\right) \\ =& \frac{1}{n}\sum_{i=1}^{n}\sum_{j:r_q^j<r_q^i} |r_q^i - r_q^j| \mathbb{I}_{\hat{r}_q^i<\hat{r}_q^j} + C_1 + C_2 \\ (*) \leq& \frac{1}{n}\sum_{i=1}^{n}\sum_{j:r_q^j<r_q^i} |r_q^i - r_q^j| \log\left(1+e^{-(\hat{r}_q^i-\hat{r}_q^j)}\right) + C_1 + C_2 \end{aligned} \tag{3}$$

where $C_1 = \frac{1}{2n}\sum_{i=1}^{n}\sum_{j:r_q^i=r_q^j} r_q^j$ and $C_2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{j:r_q^j<r_q^i} r_q^j$ ($C_2 = 0$ in binary case) are constants. As shown in the step $*$, we should only impose pairwise loss to bound the comparison indicator $\mathbb{I}_{\hat{r}_q^i<\hat{r}_q^j}$ on document pairs with different ground-truth relevance labels, for optimizing the metric (i.e., relevant-irrelevant pairs in this binary case). Comparisons from pairs with the same label form a constant C_1 that does not contribute to the optimization.

More concretely, in C_1, a pair of documents with the same label r count $r\mathbb{I}_{\hat{r}_q^i<\hat{r}_q^j} + r\mathbb{I}_{\hat{r}_q^j<\hat{r}_q^i} = r$ in the metric, independent of how they are ranked in π_q. If we introduce pairwise loss on such pairs, we are forcing the pair of documents to have the same predicted relevance values. Using pairwise logistic loss as an example, the loss $r\cdot\left[\log\left(1+e^{-(\hat{r}_q^i-\hat{r}_q^j)}\right)+\log\left(1+e^{-(\hat{r}_q^j-\hat{r}_q^i)}\right)\right]$ is minimized only when $\hat{r}_q^i = \hat{r}_q^j$. In other words, the loss on a pair of documents with the same label impose unnecessary constraints that distort the optimization.

Eq (3) reveals the root cause to the issue of IPS in practical ranker optimization using implicit feedback. We are now prepared to present our general deficiency diagnosis of existing IPS-based LTR solutions. Consider a general pairwise loss $\delta(x_i, x_j|\pi_q)$ defined on two different documents x_i and x_j in π_q, which indicates how likely x_i is more relevant than x_j. Following the IPS estimator in Eq (2), the individual contribution of a relevant document x_i to the ranking metric is thus upper bounded by the total pairwise loss of comparing x_i to all other documents in π_q:

$$\Delta(x_i|\pi_q) \leq \sum_{x_j\in\pi_q \wedge x_j\neq x_i} \delta(x_i, x_j|\pi_q) \tag{4}$$

where we assume minimizing the loss will optimize the metric.

In click data, where only the relevant and observed documents are indicated by clicks, the local loss on q can be derived from Eq (2) with pairwise loss. There are currently two strategies for imposing the pairwise loss. The first one is to compare each clicked document to *all* other documents as in Propensity SVM-Rank [19] or its generalizations [1]; but in expectation, it contains loss on all relevant-relevant pairs. To avoid explicitly comparing two relevant documents indicated by clicks, the second strategy restricts the loss on comparisons between clicked and non-clicked ones [13, 22]:

$$l_{IPS}(\pi_q|q,\tilde{\pi}_q,o_q) = \sum_{x_i:c_q(x_i)=1} \frac{\sum_{x_j:c_q(x_j)=0}\delta(x_i,x_j|\pi_q)}{P(o_q(x_i)=1|\tilde{\pi}_q)} \tag{5}$$

But as relevant and unobserved documents are also non-clicked, this strategy still cannot address the problem of including loss on relevant-relevant pairs that only distracts the optimization:

$$\begin{aligned} & E_{o_q}[l_{IPS}(\pi_q|q,\tilde{\pi}_q,o_q)] \\ =& \sum_{x_i:r_q(x_i)=1} E_{o_q}\left[\frac{o_q(x_i)\cdot\sum_{x_j:c_q(x_j)=0}\delta(x_i,x_j|\pi_q)}{P(o_q(x_i)=1|\tilde{\pi}_q)}\right] \\ =& \sum_{x_i:r_q(x_i)=1} \frac{P(o_q(x_i)=1|\tilde{\pi}_q)\cdot\sum_{x_j:c_q(x_j)=0}\delta(x_i,x_j|\pi_q)}{P(o_q(x_i)=1|\tilde{\pi}_q)} \\ =& \sum_{x_i:r_q(x_i)=1}\left(\sum_{x_j\in\pi_q:r_q(x_j)=0} + \sum_{\substack{x_j:o_q(x_j)=0 \\ \wedge r_q(x_j)=1}}\right)\delta(x_i,x_j|\pi_q) \end{aligned} \tag{6}$$

In expectation, besides the pairwise loss on relevant-irrelevant pairs that we need, this strategy also includes pairs on relevant documents versus relevant but unobserved documents (the second sum in the bracket). The negative effect can also be perceived from an intuitive perspective: as the relevant but unobserved documents are mistakenly used as negative examples, it will degrade the new ranker's ability to recognize these missed relevant documents, making the ranker still unfavorably biased against them.

In order to verify our theoretical analysis, in Figure 2, we demonstrate the empirical influence of including pairwise loss on relevant-relevant pairs in LTR model training with clicks. Specifically, we synthesize clicks on the Yahoo and Web10K LTR datasets with the propensity model ($\eta = 1$) described in Section 5. To better illustrate the effect, we adopt the noise-free setting and sampled 128,000

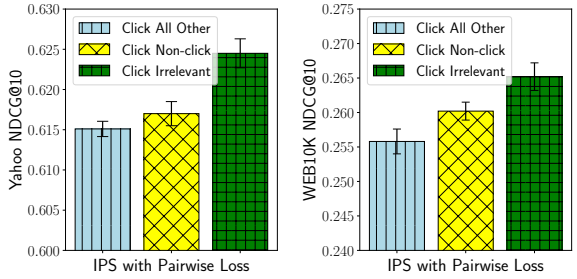

Figure 2: Empirical performance of applying IPS with pairwise loss. We compare the results of including pairwise loss on each clicked document against (1) all other documents; (2) non-clicked documents; (3) only irrelevant documents.

clicks. To learn new rankers from the clicks, we apply IPS on the pairwise logistic loss, and include pairs from different comparison strategies. The NDCG@10 results are reported on fully labeled test sets in the two benchmarks accordingly. As clearly shown, including relevant-relevant document comparisons seriously hurt ranker optimization. Therefore, we should remove the relevant-relevant pairs to eliminate such adverse effects for LTR model learning, which will be the focus of next section.

4 PROPENSITY RATIO SCORING

We have shown that to effectively tackle unbiased LTR in practice, we need to eliminate the relevant documents in non-clicked ones when counting loss, and thus keep the total loss unbiased for all relevant documents. In this section, we first derive a solution of identifying truly irrelevant documents from non-clicked ones, so as to avoid using relevant documents as negative examples to the best extent. Then we develop a holistic treatment on using click data for optimizing the ranking metric with pairwise comparisons.

4.1 Propensity-weighted Negative Samples

The insight of identifying truly irrelevant documents in non-clicked documents comes from the decomposition of click data in Figure 1. In a noise-free setting, if a document is observed (the first column in Figure 1), click is equivalent to relevance, i.e., $o_q(x_i) = 1 \rightarrow [c_q(x_i) = r_q(x_i)]$. Consequently, if a document is observed and non-clicked, it is irrelevant, i.e., $[o_q(x_i) = 1 \wedge c_q(x_i) = 0] \rightarrow r_q(x_i) = 0$ (the I&O part in Figure 1). The problem of identifying truly irrelevant documents from non-clicked ones is thus reduced to finding the observed documents in the non-clicked ones.

Note that for a non-clicked document x_j, we do not have its true observation $o_q(x_j)$. However, by weighting the loss on each non-clicked document with its conditional observation probability $P(o_q(x_j) = 1|c_q(x) = 0, \tilde{\pi}_q)$, we can restrict the loss on non-clicked documents to those that are non-clicked but observed, in expectation. But this conditional probability is not easy to estimate: As shown in Figure 1, this probability corresponds to the ratio of the *I&O* part in the non-clicked documents, which depends on both the position and relevance of the documents. Therefore, we need to estimate $P(o_q(x_j) = 1|c_q(x) = 0, \tilde{\pi}_q)$ on all positions under every single query. Without sufficient observations under the same query, the estimation quality can hardly be guaranteed. Instead, we propose to directly use the position-based observation propensity $P(o_q(x_j) = 1|\tilde{\pi}_q)$ as an approximation for the purpose of reweighing the loss on each non-clicked document,

$$\begin{aligned}
&\sum_{x_j:c_q(x_j)=0} \Omega(x_j|\pi_q) \cdot P(o_q(x_j) = 1|\tilde{\pi}_q) \qquad (7)\\
&= \Big(\sum_{x_j:r_q(x_j)=0} + \sum_{\substack{x_j:r_q(x_j)=1\\ \wedge o_q(x_j)=0}} \Big) \Omega(x_j|\pi_q) \cdot P(o_q(x_j) = 1|\tilde{\pi}_q)\\
&= E_{o_q}\Big[\sum_{\substack{x_j:r_q(x_j)=0\\ \wedge o_q(x_j)=1}} \Omega(x_j|\pi_q) \Big] + \sum_{\substack{x_j:r_q(x_j)=1\\ \wedge o_q(x_j)=0}} \Omega(x_j|\pi_q) \cdot P(o_q(x_j) = 1|\tilde{\pi}_q),
\end{aligned}$$

where $\Omega(\cdot)$ represents any general loss on a non-clicked document x_j, including the pairwise loss used in Section 3.2. Due to the approximation, the second term of Eq (7) still contains relevant documents; but $P(o_q(x_j) = 1|\tilde{\pi}_q)$ for relevant but non-clicked documents is expected to be small. For example, under a position-based examination model [19], the examination probability shrinks fast over positions; and thus the impact from relevant but non-clicked documents can be quickly eliminated when moving down the ranked list. We refer to this weighting on non-cliked documents as the Propensity-weighted Negative Samples (PNS).

4.2 Debiasing Pairwise Comparisons on Clicks

As the observation of documents are position-based and independent [10, 30], treatments on clicked and non-clicked documents will not interfere with each other. Therefore, we can integrate the inverse propensity weighting on the clciked documents and propensity weighting on the non-clicked ones to reweigh the pairwise losses incurred when comparing them:

$$\begin{aligned}
&l_{PRS}(\pi_q|q, \tilde{\pi}_q, o_q) \qquad (8)\\
&= \sum_{x_i:c_q(x_i)=1} \sum_{x_j:c_q(x_j)=0} \delta(x_i, x_j|\pi_q) \cdot \frac{P(o_q(x_j) = 1|\tilde{\pi}_q)}{P(o_q(x_i) = 1|\tilde{\pi}_q)},
\end{aligned}$$

which leads to a weight on each pairwise loss term defined by the propensity ratio between the non-clicked and clicked documents in it. We name this new weighting scheme Propensity Ratio Scoring, or PRS in short. In expectation, the PRS estimator largely removes the relevant-relevant comparisons, and focus on comparing each relevant document to irrelevant ones for ranker optimization:

$$\begin{aligned}
&E_{o_q}\big[l_{PRS}(\pi_q|q, \tilde{\pi}_q, o_q)\big] \qquad (9)\\
&= E_{o_q}\Bigg[\sum_{\substack{x_i:o_q(x_i)=1\\ \wedge\ r_q(x_i)=1}} \frac{\sum_{x_j:c_q(x_j)=0} \delta(x_i, x_j|\pi_q) \cdot P(o_q(x_j) = 1|\tilde{\pi}_q)}{P(o_q(x_i) = 1|\tilde{\pi}_q)} \Bigg]\\
&= \sum_{x_i:r_q(x_i)=1} E_{o_q}\Big[\sum_{x_j:r_q(x_j)=0, o_q(x_j)=1} \delta(x_i, x_j|\pi_q) \Big]\\
&\quad + \sum_{x_i:r_q(x_i)=1} \sum_{x_j:r_q(x_j)=1, o_q(x_j)=0} \delta(x_i, x_j|\pi_q) \cdot P(o_q(x_j) = 1|\tilde{\pi}_q),
\end{aligned}$$

where the first step is the same as Eq (6); and the second step is derived from Eq (7) by substituting $\Omega(x_j|\pi_q)$ with $\delta(x_i, x_j|\pi_q)$. As discussed in Section 4.1, the second term that consists relevant-relevant comparisons is small and can be safely ignored in practice.

In this way, the total loss is merely obtained from valid relevant-irrelevant pairs that contribute to ranker optimization. Notice that besides removing relevant documents from non-clicked documents, Eq (9) also excludes the irrelevant and unobserved documents, i.e., the I&U part in Figure 1. But as the ranking metrics are defined on relevant documents, we only need to correct the errors introduced by including relevant documents as negative examples, and keep the total loss unbiased to all relevant documents. As demonstrated in the expectation, PRS can properly promote all the relevant documents against irrelevant ones.

Besides the correction effect, we next show that by imposing a more careful use of the non-clicked documents, the variability of the PRS estimator can be reduced comparing to the IPS estimator, which is very important for LTR with real-world click data [27]. Let us first rewrite the PRS estimator in Eq (8) as follows:

$$l_{PRS}(\pi_q|q,\tilde{\pi}_q,o_q) = \sum_{\substack{x_i:o_q(x_i)=1 \\ \wedge r_q(x_i)=1}} \frac{1}{P(o_q(x_i)=1|\tilde{\pi}_q)} \cdot E_{o_q}\Big[\sum_{\substack{x_j:r_q(x_j)=0 \\ \wedge o_q(x_j)=1}} \delta(x_i,x_j|\pi_q)\Big]$$

Proposition 4.1. *Let N be the number of relevant documents retrieved for query q and $P(o_q(x_i)=1|\tilde{\pi}_q)$ be the probability of independent Bernoulli events of observing each relevant document. According to Hoeffding's inequality, for any given new ranking π_q, with probability of at least $1-\xi$, we have:*

$$\Big|l_{PRS}(\pi_q|q,\tilde{\pi}_q,o_q) - E_{o_q}\big[l_{PRS}(\pi_q|q,\tilde{\pi}_q,o_q)\big]\Big| \le \frac{1}{N}\sqrt{\frac{log\frac{2}{\xi}}{2}\sum_{i=1}^{N}\rho_i^2}$$

where $\rho_i = \frac{1}{P(o_q(x_i)=1|\tilde{\pi}_q)} \cdot E_{o_q}\big[\sum_{x_j:r_q(x_j)=0 \wedge o_q(x_j)=1} \delta(x_i,x_j|\pi_q)\big]$ *when* $0 < P(o_q(x_i)=1|\tilde{\pi}_q) < 1$*; otherwise,* $\rho_i = 0$.

The complete proof will be elaborated with details in a longer version of this paper. The above tail bound of the PRS estimator depicts its variability. Intuitively, this tail bound provides the range that the estimator can vary with a high probability; and a smaller range means a lower variability. Similarly, we can get the tail bound of the IPS estimator in Eq (5) as:

$$\Big|l_{IPS}(\pi_q|q,\tilde{\pi}_q,o_q) - E_{o_q}\big[l_{IPS}(\pi_q|q,\tilde{\pi}_q,o_q)\big]\Big| \le \frac{1}{N}\sqrt{\frac{log\frac{2}{\xi}}{2}\sum_{i=1}^{N}\tau_i^2}$$

where $\tau_i = \frac{1}{P(o_q(x_i)=1|\tilde{\pi}_q)} \cdot \sum_{x_j:c_q(x_j)=0} \delta(x_i,x_j|\pi_q)$ if $0 < P(o_q(x_i)=1|\tilde{\pi}_q) < 1$; and $\tau_i = 0$, otherwise. As the propensity of each non-clicked document is smaller than 1, $0 \le \rho_i \le \tau_i$ always holds. Thus, the PRS estimator enjoys a reduced variability than IPS, which is vital for the convergency of ranker estimation in practice.

In addition to providing an unbiased estimate of pairwise comparisons on click data, another obvious advantage of PRS is its general applicability: it does not require any additional statistics or procedures than those already used in IPS (e.g., we can use any existing methods for propensity estimation [2, 10, 19, 20, 30]). As a result, it can be seamlessly applied to all existing unbiased LTR settings or other scenarios (e.g., recommendation [25]) where IPS is used, and guaranteed for better performance.

So far, we have assumed a noise-free setting, i.e., a document is clicked if and only if it is observed and relevant: $c_q(x_i) = 1 \Leftrightarrow [o_q(x_i) = 1 \wedge r_q(x_i) = 1]$. However, in reality this may not hold: a user can possibly misjudge and miss a relevant document, or mistakenly click on an irrelevant document. Fortunately, PRS is order-preserving under the same noise assumption made by IPS [19]. Due to space limit, we decide not to include the proof, which can be obtained similarly as in [19] but based on Eq (9). We will present the influence of noisy clicks empirically in Section 5.3.

5 EVALUATION

In this section, we conduct comprehensive empirical evaluations of PRS for unbiased LTR. We apply PRS to different ranking algorithms to show its wide applicability. We first synthesize the clicks following the conventional procedure [3, 13, 19] on three benchmark LTR datasets to study the behaviors of PRS from different perspectives. To confirm the effectiveness of PRS in an industrial setting, we also perform experiments on the large-scale GMail search data.

5.1 PRS on Different Ranking Models

As shown in Eq (8), the proposed PRS estimator generally applies to any pairwise LTR algorithms. To test the performance of PRS on different ranking models, we include two popularly used yet significantly different pairwise LTR algorithms for experiments. One is pairwise logistic regression [4, 22], and the other one is LambdaMART [5], the state-of-the-art pairwise LTR algorithm.

• **Pairwise Logistic Regression.** It uses a logistic function to measure the likelihood of x_i being more relevant than x_j under query q, based on their predicted relevance. By taking the logarithm of the logistic function, the pairwise logistic loss for optimization is:

$$\delta(x_i,x_j|\pi_q) = \log\left(1+e^{-\left(\tilde{r}_q(x_i)-\tilde{r}_q(x_j)\right)}\right)$$

We use a linear scoring function $\tilde{r}_q(x_i) = \omega^\top \phi(x_i,q)$, where $\phi(x_i,q)$ is a feature vector that describes the matching between x_i and q. We can directly substitute the above loss in Eq (8) to get the PRS estimator for pairwise logistic regression and add an l_2 regularization to control overfitting.

• **LambdaMART.** It combines MART [12] and the lambda functions from LambdaRank [6]. Its optimization is directly performed with respect to the lambda functions. To apply PRS with click data in LambdaMART, we first denote a set of document pairs $I_q = \{(x_i,x_j)|c_q(x_i)=1 \wedge c_q(x_j)=0\}$ in each query q. Then we modify the lambda functions as:

$$\tilde{\lambda}_i = \sum_{j:(x_i,x_j)\in I_q} \tilde{\lambda}_{ij} - \sum_{j:(x_j,x_i)\in I_q} \tilde{\lambda}_{ji}$$

$$\text{where } \tilde{\lambda}_{ij} = \frac{-\sigma|\Delta Z_{ij}|}{1+e^{\sigma(\tilde{r}_q(x_i)-\tilde{r}_q(x_j))}} \cdot \frac{P(o_q(x_j)=1|\tilde{\pi}_q)}{P(o_q(x_i)=1|\tilde{\pi}_q)}$$

ΔZ_{ij} is the change of the ranking metric of interest (e.g., NDCG) if documents x_i and x_j are swapped in the ranked list π_q; σ is a global weighting coefficient. $\tilde{\lambda}_{ij}$ without the PRS weight can be viewed as the pairwise loss $\delta(x_i,x_j|\pi_q)$ in Eq (8). We applied the modified lambda functions in the standard LambdaMART implementation, which we name as PRS-LambdaMART.

We primarily compare the PRS estimator against the IPS estimator, and also a Naive estimator. The Naive estimator simply treats clicks as relevance and non-clicks as irrelevance. The two baseline estimators can be viewed as special cases of our PRS estimator: IPS

always sets the observation propensity of a non-clicked document to 1; and Naive simply sets the PRS weight to 1 in all pairs. All three estimators are applied to the two base ranking algorithms to learn new rankers from click data. We also include the Full-Info ranker trained with fully-labeled ground-truth data as the skyline.

5.1.1 Reasoning of Weight Clipping. Similar to the clipping trick used in IPS [19], proper clipping on the PRS weights is important to safeguard its stable performance in real applications. The main reason of weight clipping is for variance reduction, and we found the variance highly depends on the quality of the ranker that presents the logged ranking $\tilde{\pi}_q$ (referred to as production ranker in [19]). Specifically, when a low-quality production ranker is used when collecting clicks, the chance that some irrelevant documents ranked at the top and relevant documents ranked at the bottom will be high. Hence, there will be pairs in which a bottom ranked relevant document is clicked and a top ranked irrelevant document is not. As the PRS weight is the ratio between the propensities of the non-clicked and clicked documents, such pairs will have very large PRS weights and thus dominate the total loss. The resulting ranker will then be forced to correct these edge cases, e.g., go against or even reverse the production ranker, but totally miss other documents. In practice, to avoid such effect, we clip the PRS weights in Eq (8) with a constant γ (only clipping the small propensities of lower-ranked clicked documents as in IPS [19] yields similar effect):

$$l_{PRS}(\pi_q|q,\tilde{\pi}_q,o_q) \tag{10}$$
$$= \sum_{x_i:c_q(x_i)=1} \sum_{x_j:c_q(x_j)=0} \delta(x_i,x_j|\pi_q)\cdot \min\left\{\gamma, \frac{P(o_q(x_j)=1|\tilde{\pi}_q)}{P(o_q(x_i)=1|\tilde{\pi}_q)}\right\}$$

Empirically, we set $\gamma = 1$ which is found to be effective in our experiments. However, when the ranking model used to train a new ranker has sufficient capacity (e.g., a non-linear model) to fit all pairs including both the extreme and regular ones, the influence of those large weights becomes less a concern and we can relax the clipping. For example, LambdaMART is less sensitive to the extreme pairs than Logistic Regression in our observations. For the IPS estimator, the same analysis applies and we follow the clippings suggested in [19] to make a fair comparison.

5.2 Synthesize Clicks on LTR Benchmarks

We adopt three benchmark LTR datasets, including Yahoo Learning to Rank Challenge (set1), MSLR-WEB10K and MQ2007.

• **Yahoo.** One of the largest and most popularly used benchmark dataset for LTR. It contains around $30K$ queries with $710K$ documents. Each query-document pair is depicted with a 700-dimension feature vector (519 valid features) and a five-grade relevance label. Following [19], we binarize the relevance by assigning $r_q(x) = 1$ to documents with relevance label 3 or 4, and $r_q(x) = 0$ for others.

• **WEB10K.** It contains $10K$ queries and a 136-dimension feature vector for each query-document pair. We binarize the relevance labels in the same way as in the Yahoo dataset.

• **MQ2007.** It contains about $1,700$ queries and a 46-dimension feature vector for each query-document pair, with relevance label in $\{0,1,2\}$. We assign $r_q(x) = 1$ to documents with relevance label 1 or 2, and $r_q(x) = 0$ to documents with relevance label 0.

For all three datasets, we keep the partition of the train, validation, test set from the corpus and report the performance on the binarized fully labeled test sets with five-fold cross-validation. We follow the procedure in [19] to derive click data from each fully labeled dataset. To generate clicks, we first train an initial ranker with 1 percent of the queries in the training set, which is referred to as the production ranker π_0. Then we randomly select a query q from the rest of the training set, for which we use π_0 to compute the ranking $\tilde{\pi}_q$ for this query. With the ranked list, we can generate clicks according to a position-based click model,

$$P(c_q(x_i)=1|\tilde{\pi}_q) = P(e_q(x_i,k)=1|\tilde{\pi}_q)\cdot P\big(r_q(x_i)=1|e_q(x_i,k)=1\big),$$

where $e_q(x_i,k)$ denotes whether the document at position k is examined, and examination equals the observation of the relevance label: $e_q(x_i) = o_q(x_i)$. Thus the observation propensity is equivalent to the position-based examination probability $P(e_q(x_i,k)=1|\tilde{\pi}_q)$, which is defined as follows,

$$P(o_q(x_i)=1|\tilde{\pi}_q) = p_{rank(x_i|\tilde{\pi}_q)} = \left(\frac{1}{rank(x_i|\tilde{\pi}_q)}\right)^{\eta}$$

where η represents the severity of position bias. The propensity $P(o_q(x_i)=1|\tilde{\pi}_q)$ of each document (regardless of clicked or non-clicked) is recorded based on their positions in the presented ranking. We also add click noise as in [19]. Denote $\mu \in [0, 0.5)$ as the noise level, we have $P\big(c_q(x_i)=1\big|r_q(x_i)=1, o_q(x_i)=1\big) = 1-\mu$ and $P\big(c_q(x_i)=1\big|r_q(x_i)=0, o_q(x_i)=1\big) = \mu$. When not mentioned otherwise, we use $\eta = 1$ and $\mu = 0.1$ as the default setting. In the following sections, we investigate the influence of each component.

We use NDCG@10 as the main performance metric. We also computed other metrics such as MAP and ARP. But as the performance on these metrics was consistent with each other and the space limit, we only report NDCG@10 to include more experiments. We tune the hyper-parameters via cross-validation. Each result is averaged over five runs and the standard deviation is displayed as the shadow areas in the figures.

5.3 Evaluations on Synthetic Data

5.3.1 Performance with the Scale of Click Data. We first study how the ranking performance scales with the number of clicks. The results are reported in Figure 3, and the production ranker π_0 is used as a baseline. The x-axis denotes the number of clicks, and the y-axis reports the NDCG@10 on the fully labeled test set. The figures show that PRS consistently and significantly outperforms IPS with an increasing size of the click data, in both ranking algorithms across three datasets. When there are only a few clicks, the ranking model cannot be sufficiently estimated. Hence the size of training data is the major bottleneck rather than the bias in it. But with an increasing number of clicks, IPS introduces more relevant documents from the non-clicked part into the comparisons, which distorts the optimization of rankers. Besides, the variance of PRS is reduced considerably due to a finer-grained use of the non-clicks. The Naive estimator does not consider the bias at all and thus cannot make effective use of click data. This experiment demonstrates clear advantages of PRS in our default setting. Next we will focus on specific perspectives of the unbiased LTR settings. Due to the space limit, we only report the results on Yahoo dataset, but the conclusions are consistent with the other two datasets.

5.3.2 Tolerance to the Severity of Position Bias. We investigate the performance of the PRS estimator under different degrees of

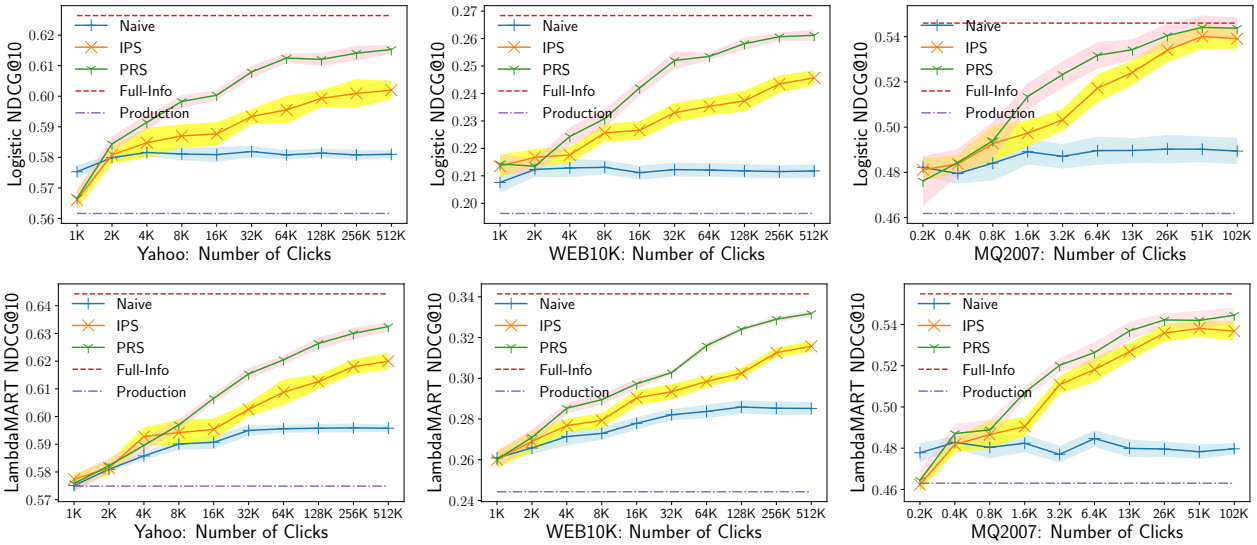

Figure 3: The test set ranking performance of different rankers. The results are from the clicks on two base ranking algorithms over three datasets as indicated in the figures. The shadow areas denote standard deviations at each result point. ($\eta = 1, \mu = 0.1$)

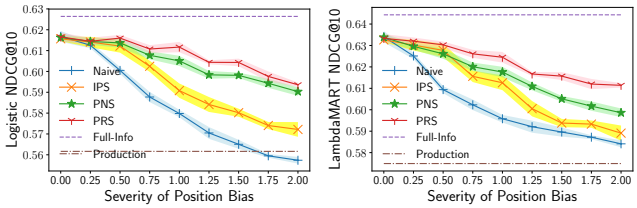

Figure 4: The performance of different rankers under different degree of position bias controlled by η. ($n = 128K, \mu = 0.1$)

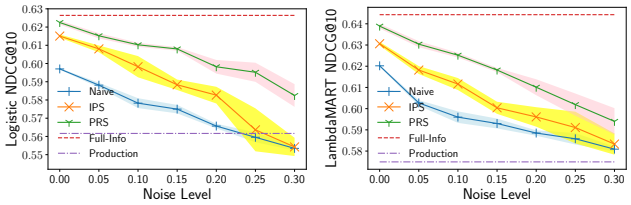

Figure 5: The performance of different rankers with an increasing click noise level ($n = 128K, \eta = 1$).

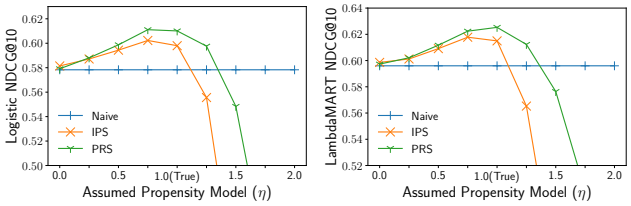

Figure 6: The performance of different rankers with misspecified propensities ($n = 128K$, true $\eta = 1$, $\mu = 0.1$).

position bias in clicks. We vary η from 0 to 2 when generating clicks. In order to better understand the effect of position bias, we also include the weighting solely on non-clicked documents (PNS) as introduced in Section 4.1. PNS focuses on identifying truly irrelevant documents in unclicked documents, but does not correct the bias on clicks. Figure 4 demonstrates the influence of the bias on different estimators for LTR. PRS achieves the best performance as it properly handles both clicks and non-clicks. The IPS estimator only works when there is a low level of position bias, while its advantage over the Naive estimator diminishes with increased bias. This is consistent with the previously reported findings in [15]. It is worth noting that the PNS estimator is more robust to stronger position bias than IPS and Naive. This observation suggests that the consequence of including relevant documents as negative examples can be even more severe than the bias issue in clicked documents.

5.3.3 Robustness to Click Noise. We now evaluate the robustness of different estimators to click noise, by varying the noise level μ in Figure 5. The results show that PRS is more resistant to click noise. This is again due to its treatments on both clicked and non-clicked documents. For example, when an irrelevant document is clicked, its large IPS weight will be canceled by the propensity weight on the non-clicked document introduced by PRS. And therefore, this erroneous pair generates less impact on ranker estimation. In comparison, when the noise level is high, the performance of IPS can drop to the Naive estimator's performance, because of its unreasonably high weight on the erroneous pairs.

5.3.4 Robustness to Misspecified Propensities. We have assumed the access to accurate propensities. However, this is not always the case in practice and the propensities may need to be estimated based on a specified propensity model. In this experiment, we evaluate how robust different estimators are to misspecified propensities. We use $\eta = 1$ to generate clicks, but with different η in training. The results are shown in Figure 6, where x-axis is the η for the assumed (estimated) propensities in click data. Both IPS and PRS are less sensitive to the overestimated propensities (when $\eta < 1$). But PRS is much more robust than IPS when the propensities are

Table 1: The ranking performance of different unbiased LTR solutions ($n = 128K, \eta = 1, \mu = 0.1$).

	MAP	NDCG@5	NDCG@10
Propensity SVM-Rank	0.5720	0.5877	0.5978
DLA	0.5785	0.5974	0.6009
Unbiased LambdaMART	0.5961	0.6121	0.6173
PRS-LambdaMART	**0.6049***	**0.6206***	**0.6264***

underestimated (when $\eta > 1$). This is because in PRS, as long as the propensity *ratios* are close to those from the true propensities, the ranker estimation quality could be largely maintained. This analysis also illustrates the practical advantage of PRS when accurate propensities are difficult to obtain.

5.3.5 Comparisons with Variants of IPS. We use PRS-LambdaMART as our solution for unbiased LTR, and compare it with recent variants of IPS-based methods. The first one we include is Propensity SVM-Rank [19], which applies IPS to SVM-Rank. The second baseline is the Dual Learning Algorithm (DLA) [3]. It uses a DNN model to jointly estimate propensities and an unbiased ranker from click data as a dual problem. The last one is Unbiased LambdaMART [13], which applies IPS weights to both clicked and non-clicked documents and jointly estimates the propensities and the ranker in a pairwise manner. For a fair comparison, we estimate the propensities with randomization from the generated clicks as used in [3, 13] for our solution. The comparisons of the ranking performance are shown in Table 1 under paired t-test with p-value<0.05. First, pairwise comparisons between clicked and non-clicked documents used in Unbiased LambdaMART and PRS-LambdaMART are more effective than comparing each clicked documents to all others as in the other two baselines. The Unbiased LambdaMART algorithm can be understood as a special case of our PRS scheme. It applies the IPS weights to non-clicked documents as $\frac{1}{t^-}$, by assuming the non-click probability is proportional to the irrelevance probability. As t^- is not bounded, i.e., it can be larger than 1, $\frac{1}{t^-}$ could achieve a similar effect as the propensity weight on non-clicked documents. But as t^- is not bounded and largely relies on the regularization for estimation, there is no guarantee it can recover the correct propensity and achieve the desired effect. This leads to its worse performance. On the other hand, this also shows the possibility of generalizing PRS to jointly learning propensities and the unbiased ranker.

5.4 Evaluations on GMail Search Data

We further evaluate PRS on data from one of the world's largest personal search engines, GMail search, to prove PRS' applicability in real-world large-scale industrial settings. Specifically, we collected the click-through data from Gmail search logs between June and July 2020 for experiments, resulting in hundreds of millions of queries with clicks. The ranking order of documents in each query was determined by the actual deployed production model, and other factors such as (the unknown) click noise are without any synthetic interventions. Among all the queries, 80% are used for training, 10% for validation and parameter tuning, and the remaining for testing. On average, each GMail query has six candidate documents based on its search interfaces and one of the documents is clicked.

The dataset contains a rich set of features, including query-document matching features such as BM25, situational features (e.g. time of the day), user features (e.g. user's previous click behavior), and document attributes (e.g. document age), etc. The observation propensities are estimated from randomized online experiments [29]. We use weighted mean reciprocal rank (WMRR) as our primary metric, since it has been found to be predictive of online gains [30]. In particular, WMRR is calculated as follows:

$$\text{WMRR} = \frac{1}{\sum_{i=1}^{N} w_i} \cdot \sum_{i=1}^{N} w_i \frac{1}{\text{rank}_i}, \tag{11}$$

where w_i denotes the bias correction weight that corresponds to the inverse propensity of the clicked document, N denotes the number of testing queries, rank_i denotes the rank position of the clicked document for the i-th query. As explained in Section 3.1, WMRR is an unbiased estimate of MRR metric. Hence, the higher WMRR is, the better a model performs.

Table 2: Ranking performance on GMail data using WMRR.

	Overall	Clicked at top	Clicked at others
IPS-linear	+0.11%	-0.19%	+0.40%
PRS-linear	+0.39%	+0.13%	+0.61%
Unbiased LambdaMART	+3.92%	+5.54%	+0.77%
PRS-LambdaMART	+4.14%	+5.71%	+0.99%
IPS-DNN	+3.91%	+5.65%	+0.64%
PRS-DNN	+4.11%	+5.69%	+0.96%

We conduct experiments with linear rankers, LambdaMART, and deep neural network (DNN) rankers to demonstrate PRS' robust performance on real-world datasets with different families of ranking models. We include the following specific models for comparison: a baseline logistic regression model using Naive estimator, an IPS-trained logistic regression, a PRS-trained logistic regression, Unbiased LambdaMART (based on IPS) [13], PRS-LambdaMART, an IPS-trained DNN model, and a PRS-trained DNN model. For all compared models, we use the same set of input features. For DNN models, we use a 3-layer fully connected architecture with 256 hidden dimensions, pairwise logistic loss, and SGD for optimization. All results are reported comparatively to the base logistic regression model, to avoid reporting the absolute performance that is proprietary. As a large portion of testing queries are already clicked at the top (position 0), which are considered as easier queries, we also report the results on other queries separately to better illustrate the performance improvement. The results are shown in Table 2. Note that in large commercial search systems, an approach with an improvement around 0.2% is considered as substantial [21, 29].

All relative differences between PRS and baseline in each family of algorithms are statistically significant under paired t-test with p-value<0.05. We can find that PRS not only performs the best among all solutions, it is also robust under different queries, including those more difficult ones (i.e., clicked on lower ranked positions). IPS tends to over-emphasize difficult queries (e.g., give them larger weights in training), while sacrificing performance on easier ones. LambdaMART and DNN rankers outperform linear rankers by a large margin due to their model capacity in leveraging feature non-linearity. Unbiased LambdaMART shows competitive performance, but is still inferior to PRS-LambdaMART on this large-scale real-world dataset. The results also strongly support PRS for real-world deployments: it only requires a simple modification of the instance

weighting schema during training, without any other changes (e.g., logging or optimization).

6 CONCLUSIONS AND FUTURE WORK

In this work, we identify and prove the deficiency of IPS in unbiased LTR tasks. In particular, IPS inevitably includes the relevant-relevant document comparisons when using click data for LTR, which distorts ranker optimization. We instead propose a new weighting scheme named PRS that imposes treatments on both clicks and non-clicks. Comprehensive empirical evaluations on both synthetic and real-world Gmail search data demonstrate the significance of PRS for practical use.

This paper lays the theoretical basis of effectively utilizing implicit feedback, such as clicks, for LTR. The proposed PRS solution can be easily applied to where IPS is used without any infrastructure change. Although PRS is developed based on pairwise comparisons, it is straightforward to generalize to pointwise LTR methods. Besides, recent advances in listwise LTR also indicate the importance of comparing each document to its less relevant peers [34]. It is important to extend the idea of removing relevant documents from non-clicks in PRS for effective listwise LTR.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful comments and suggestions. This work is partially supported by the National Science Foundation under grant SCH-1838615, IIS-1553568, and a Google Faculty Research Award.

REFERENCES

[1] Aman Agarwal, Kenta Takatsu, Ivan Zaitsev, and Thorsten Joachims. 2019. A General Framework for Counterfactual Learning-to-Rank. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. ACM, New York, NY, USA, 5–14.

[2] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. 2019. Estimating Position Bias Without Intrusive Interventions. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19)*. ACM, New York, NY, USA, 474–482.

[3] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W. Bruce Croft. 2018. Unbiased Learning to Rank with Unbiased Propensity Estimation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. ACM, New York, NY, USA, 385–394.

[4] J. P. Arias-Nicolás, C. J. Pérez, and J. Martín. 2008. A logistic regression-based pairwise comparison method to aggregate preferences. *Group Decision and Negotiation* 17, 3 (01 May 2008), 237–247.

[5] Chris J.C. Burges. 2010. *From RankNet to LambdaRank to LambdaMART: An Overview*. Technical Report MSR-TR-2010-82.

[6] Christopher J. C. Burges, Robert Ragno, and Quoc Viet Le. 2006. Learning to Rank with Nonsmooth Cost Functions. In *Proceedings of the 19th International Conference on Neural Information Processing Systems (NIPS'06)*. MIT Press, Cambridge, MA, USA, 193–200.

[7] Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. 2012. Large-Scale Validation and Analysis of Interleaved Search Evaluation. *ACM Trans. Inf. Syst.* 30, 1, Article Article 6 (March 2012), 41 pages.

[8] Olivier Chapelle and Ya Zhang. 2009. A Dynamic Bayesian Network Click Model for Web Search Ranking. In *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*. ACM, 1–10.

[9] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2016. Click Models for Web Search and Their Applications to IR: WSDM 2016 Tutorial. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (WSDM '16)*. ACM, 689–690.

[10] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An Experimental Comparison of Click Position-Bias Models. In *Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM '08)*. ACM, 87–94.

[11] Georges E. Dupret and Benjamin Piwowarski. 2008. A User Browsing Model to Predict Search Engine Click Data from Past Observations.. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*. ACM.

[12] Jerome H. Friedman. 2001. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics* 29, 5 (2001), 1189–1232.

[13] Ziniu Hu, Yang Wang, Qu Peng, and Hang Li. 2019. Unbiased LambdaMART: An Unbiased Pairwise Learning-to-Rank Algorithm. In *The World Wide Web Conference (WWW '19)*. ACM, New York, NY, USA, 2830–2836.

[14] Guido W. Imbens and Donald B. Rubin. 2015. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction.* Cambridge University Press, USA.

[15] Rolf Jagerman, Harrie Oosterhuis, and Maarten de Rijke. 2019. To Model or to Intervene: A Comparison of Counterfactual and Online Learning to Rank from User Interactions. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. ACM, 15–24.

[16] Thorsten Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*. ACM, New York, NY, USA, 133–142.

[17] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2017. Accurately Interpreting Clickthrough Data as Implicit Feedback. *SIGIR Forum* 51, 1 (Aug. 2017), 4–11.

[18] Thorsten Joachims, Laura 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 Trans. Inf. Syst.* (April 2007).

[19] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM '17)*. ACM, New York, NY, USA, 781–789.

[20] Zhen Qin, Suming J. Chen, Donald Metzler, Yongwoo Noh, Jingzheng Qin, and Xuanhui Wang. 2020. Attribute-Based Propensity for Unbiased Learning in Recommender Systems: Algorithm and Case Studies. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '20)*. 2359–2367.

[21] Zhen Qin, Zhongliang Li, Michael Bendersky, and Donald Metzler. 2020. Matching Cross Network for Learning to Rank in Personal Search. In *Proceedings of The Web Conference (WWW '20)*. 2835–2841.

[22] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09)*. AUAI Press, Arlington, Virginia, USA, 452–461.

[23] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting Clicks: Estimating the Click-through Rate for New Ads. In *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*. ACM, 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 *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48 (ICML'16)*. JMLR.org, 1670–1679.

[26] Adith Swaminathan and Thorsten Joachims. 2015. Batch Learning from Logged Bandit Feedback through Counterfactual Risk Minimization. *Journal of Machine Learning Research* 16, 52 (2015).

[27] Adith Swaminathan and Thorsten Joachims. 2015. The Self-Normalized Estimator for Counterfactual Learning. In *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 3231–3239.

[28] Hongning Wang, ChengXiang Zhai, Anlei Dong, and Yi Chang. 2013. Content-Aware Click Modeling. In *Proceedings of the 22nd International Conference on World Wide Web (WWW '13)*. ACM, 1365–1376.

[29] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to Rank with Selection Bias in Personal Search. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)*. ACM.

[30] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position Bias Estimation for Unbiased Learning to Rank in Personal Search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. ACM, New York, NY, USA, 610–618.

[31] Xuanhui Wang, Cheng Li, Nadav Golbandi, Michael Bendersky, and Marc Najork. 2018. The LambdaLoss Framework for Ranking Metric Optimization. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. ACM, New York, NY, USA, 1313–1322.

[32] Yisong Yue and Thorsten Joachims. 2009. Interactively Optimizing Information Retrieval Systems as a Dueling Bandits Problem. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*. ACM, 1201–1208.

[33] Yisong Yue, Rajan Patel, and Hein Roehrig. 2010. Beyond Position Bias: Examining Result Attractiveness as a Source of Presentation Bias in Clickthrough Data. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*. ACM, 1011–1018.

[34] Xiaofeng Zhu and Diego Klabjan. 2020. Listwise Learning to Rank by Exploring Unique Ratings. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM '20)*. ACM, 798–806.