When Inverse Propensity Scoring does not Work: Affine Corrections for Unbiased Learning to Rank

Vardasbi, A.; Oosterhuis, H.; de Rijke, M.

DOI
10.1145/3340531.3412031
Publication date
2020

Published in
CIKM '20

When Inverse Propensity Scoring does not Work: Affine Corrections for Unbiased Learning to Rank

Ali Vardasbi*
University of Amsterdam
Amsterdam, The Netherlands
a.vardasbi@uva.nl

Harrie Oosterhuis*
University of Amsterdam
Amsterdam, The Netherlands
oosterhuis@uva.nl

Maarten de Rijke
University of Amsterdam & Ahold Delhaize
Amsterdam, The Netherlands
derijke@uva.nl

ABSTRACT

Besides position bias, which has been well-studied, trust bias is another type of bias prevalent in user interactions with rankings: users are more likely to click incorrectly w.r.t. their preferences on highly ranked items because they trust the ranking system. While previous work has observed this behavior in users, we prove that existing Counterfactual Learning to Rank (CLTR) methods do not remove this bias, including methods specifically designed to mitigate this type of bias. Moreover, we prove that Inverse Propensity Scoring (IPS) is principally unable to correct for trust bias under non-trivial circumstances. Our main contribution is a new estimator based on affine corrections: it both reweights clicks and penalizes items displayed on ranks with high trust bias. Our estimator is the first estimator that is proven to remove the effect of both trust bias and position bias. Furthermore, we show that our estimator is a generalization of the existing CLTR framework: if no trust bias is present, it reduces to the original IPS estimator. Our semi-synthetic experiments indicate that by removing the effect of trust bias in addition to position bias, CLTR can approximate the optimal ranking system even closer than previously possible.

KEYWORDS

Unbiased learning to rank; Inverse propensity scoring; Position bias; Trust bias

ACM Reference Format:
Ali Vardasbi, Harrie Oosterhuis, and Maarten de Rijke. 2020. When Inverse Propensity Scoring does not Work: Affine Corrections for Unbiased Learning to Rank. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3412031

1 INTRODUCTION

Learning to Rank (LTR) is a long-established area of research that continues to receive considerable attention from academia and industry [14]. Supervised approaches to LTR use manually annotated data, where human annotators have provided relevance judgements. Over time, the limitations of such approaches have become apparent: manually annotated labels are time consuming and expensive to create [5, 17]; moreover, the preferences of actual users and annotators need not be aligned [18]. Instead, recent years have brought increased interest in LTR methods that learn from user interactions.

At first glance user interactions have great advantages over labelled data: online search engines receive large numbers of interactions at virtually no additional costs; and interactions reflect actual user preferences as opposed to annotators' preferences. Unfortunately, user interactions also bring their own difficulties because they are a form of noisy and biased implicit feedback. For instance, clicks are noisy in the sense that, often, a non-relevant item receives a click or a relevant item is skipped. The effect of noise is easily mitigated by averaging over a large number of clicks, but this is not true for bias. *Position bias*, a well-known type of bias of interactions through clicks [6], occurs because users are more likely to examine results at higher ranks. As a consequence, an item may receive more clicks because it was displayed at a high rank, not because it was preferred by the user. Other types of bias include *item-selection bias*: not all items can be displayed at once [15, 16]; *presentation bias*: items are presented in different manners [23]; and *trust-bias*: users are more likely to click incorrectly on higher ranked items [12]. In order to infer a user's true preferences from their interactions, the effects of these biases have to be corrected for.

Research into Counterfactual Learning to Rank (CLTR) aims to find methods that learn from user interactions but whose optimization process is unaffected by biases [13]. Early CLTR methods correct for position bias using Inverse Propensity Scoring (IPS) [13, 20]. IPS estimators weight clicks inversely to the probability of the clicked items being examined during logging. Thus, clicks on items that are less likely to have been examined by users are weighted more heavily. This reweighting compensates for the effect of position bias, allowing CLTR methods to estimate and learn without being affected by position bias in expectation. Later CLTR work has focused on estimating examination probabilities [3, 4, 8, 21], training deep learning models [1], and correcting for more types of bias [2, 15, 16]. In particular, Agarwal et al. [2] have proposed an expansion to IPS to correct for both position bias and trust bias.

In this paper, we prove that *no IPS estimator is able to correct for trust bias*, under non-trivial circumstances. Since all existing bias mitigation methods are IPS-based approaches, this implies that there is currently no known CLTR method that can deal with trust bias. We identify the root cause to be the fact that IPS only corrects for Missing-Not-At-Random (MNAR) feedback [13]. While position bias prevents clicks from occurring due to a lack of user examination, trust bias adds additional clicks due to user trust [2, 12].

* Equal contribution.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412031

Table 1: Notation used in this paper.

Symbol	Description
q	a query
d	an item (to be ranked)
D_q	set of items to be ranked for query q
λ	metric function that assigns a weight per rank
f	a ranker, or ranking function, that scores items
y_i	a ranking displayed at interaction i
C	a click on an item
E	user examination of an item
R	the relevance of an item
$\tilde{R}$	the perceived relevance of an item
θ_k	examination probability at rank k: $P(E = 1 \mid k)$
$\gamma_{q,d}$	relevance probability: $P(R = 1 \mid q, d)$
ϵ_k^+	perceived relevance probability at rank k of an examined relevant item: $P(\tilde{R} = 1 \mid E = 1, R = 1, k)$
ϵ_k^-	perceived relevance probability at rank k of an examined non-relevant item: $P(\tilde{R} = 1 \mid E = 1, R = 0, k)$
α_k	first weight of the affine transformation of trust bias: $\alpha_k = \theta_k(\epsilon_k^+ - \epsilon_k^-)$
β_k	second weight of the affine transformation of trust bias: $\beta_k = \theta_k \epsilon_k^-$

Hence, clicks that are affected by trust bias are not simply a form of MNAR feedback and IPS cannot correct for such biases.

We introduce a novel estimator for CLTR that makes use of *affine* corrections, as opposed to the *linear* corrections of IPS. Our novel affine estimator both reweights clicks based on examination probabilities and penalizes items for being displayed on ranks where many incorrect clicks take place. We prove that the affine estimator is the first method that can correct for both position bias and trust bias. Furthermore, we show that it is an extension of the existing CLTR framework: when no trust bias is present the affine estimator naturally reduces to an IPS estimator. The results of our semi-synthetic experiments show that while existing CLTR methods are negatively affected by trust bias, our affine approach approximates the optimal ranking model under varying degrees of position bias and trust bias.

The main contributions of this work are:

(1) The first CLTR estimator that is proven to be unbiased w.r.t. both position bias and trust bias.
(2) A theoretical analysis that shows that IPS estimators cannot correct for trust bias.
(3) An empirical analysis based on semi-synthetic experiments that reveal our affine estimator bridges the gap between existing CLTR methods and the optimal model when trust bias is present.

Table 1 summarizes the notation we use in the paper.

2 BACKGROUND AND RELATED WORK

This section covers supervised LTR and the original IPS method for CLTR with position bias correction.

2.1 Learning to rank

In general, the goal of LTR methods is to find the optimal ranking function f, in order to sort items for user-issued queries. For this work, we will use f to sort in ascending order. Let q indicate a query, d an item, and $\text{rank}(d \mid q, f)$ the rank of item d in the ranking produced by f for q. Then:

$$f(d_i \mid q) > f(d_j \mid q) \Rightarrow \text{rank}(d_i \mid q, f) \succ \text{rank}(d_j \mid q, f). \quad (1)$$

Commonly, f is considered optimal if it maximizes some linearly decomposable metric. Let $P(q)$ be the distribution of queries, D_q the set of items to be ranked for query q, and $P(R = 1 \mid q, d)$ the probability that an item d is considered relevant by the user. Then, with some weighting function λ, a linearly decomposable metric has the form:

$$\Delta(f) = \sum_q P(q) \sum_{d \in D_q} P(R = 1 \mid q, d) \cdot \lambda(d \mid q, f). \quad (2)$$

Generally, λ is based on the rank of d for q according to f. For instance, it can be chosen to match the well-known Discounted Cumulative Gain (DCG) metric:

$$\lambda_{\text{DCG}}(d \mid q, f) = \Big(\log_2 \big(\text{rank}(d \mid D_q, q, f) + 1\big) \Big)^{-1}. \quad (3)$$

If the relevance probabilities $P(R = 1 \mid q, d)$ are known, finding the optimal f can be done through traditional supervised LTR methods [14].

2.2 Counterfactual learning to rank for position bias correction

In practice the relevance probabilities $P(R = 1 \mid q, d)$ are not known and are costly to estimate through human labelling [5, 17]. Moreover, often the annotations obtained through manual labelling are not aligned with the actual preferences of the users [18].

As an alternative, Counterfactual Learning to Rank (CLTR) methods use click logs to base their optimization and evaluation on. Clicks can be seen as a form of implicit feedback, which is indicative of the users' preferences but also a very noisy and biased signal. One of the most prevalent biases in clicks on items included in a ranking is *position bias*: users are less likely to examine – and therefore click – items on lower ranks. Position bias is formally modeled through the examination hypothesis, which states that a clicked item ($C \in \{0, 1\}$) must be examined ($E \in \{0, 1\}$) and considered relevant ($R \in \{0, 1\}$): $C = 1 \leftrightarrow E = 1 \wedge R = 1$. Position bias is often assumed to depend on the rank at which an item is displayed, while the relevance of an item is assumed to be independent of where it is displayed [6]. Thus, if k is the rank at which d is displayed, the probability of a click is:

$$P\big(C = 1 \mid q, d, k\big) = P\big(E = 1 \mid k\big) \cdot P\big(R = 1 \mid q, d\big). \quad (4)$$

The click probability (Eq. 4) shows us that the position bias, modeled by $P(E = 1 \mid k)$, gives an unfair advantage to documents in positions that are more likely to be examined.

Let $\mathcal{D}$ be the set of logged interactions, containing N tuples each consisting of a user-issued query q_i, a displayed ranking y_i, and the observed clicks c_i where $c_i(d) \in \{0, 1\}$:

$$\mathcal{D} = \{(q_i, y_i, c_i)\}_{i=1}^N. \quad (5)$$

For brevity we will use the sum $\sum_{(d,k) \in y_i}$, which sums over the items d and their associated ranks in y_i:

$$\forall (d, k) \in y_i, \;\; k = \text{rank}(d \mid i). \quad (6)$$

Furthermore, we use $P(E = 1 \mid k) = \theta_k$. Thus, the probability of item d being examined at interaction i depends on the rank it was displayed at: $P(E = 1 \mid d, i) = \theta_{\text{rank}(d|i)}$. The first published CLTR methods correct for position bias using an IPS estimator [13, 20]. This IPS estimator weights each click inversely to the probability that the clicked item was examined:

$$\hat{\Delta}_{IPS}(f) = \frac{1}{N} \sum_{i=1}^{N} \sum_{(d,k) \in y_i} \frac{c_i(d)}{\theta_k} \cdot \lambda(d \mid q_i, f). \tag{7}$$

The result is an unbiased estimator, since in expectation it correctly estimates Δ:

$$\mathbb{E}_{q,y,c}[\hat{\Delta}_{IPS}(f)] = \Delta(f) \quad \text{(under the click model in Eq. 4)}. \tag{8}$$

For a proof of unbiasedness we refer to the work by Joachims et al. [13], who prove that even with click noise $\hat{\Delta}_{IPS}$ can be used for unbiased CLTR optimization. However, we note that this proof relies on (at least) three important assumptions: (i) the click model as described in Eq. 4 is true, (ii) the propensities θ are known, and (iii) all propensities are positive: $\forall k \ \theta_k > 0$.

A lot of related work has considered the estimation of the position bias parameters θ, using randomization [3, 8, 13, 21], or by jointly estimating relevance and position bias [4, 20]. Recently, both Ovaisi et al. [16] and Oosterhuis and de Rijke [15] have proposed using different propensities when not all items can be displayed at once (i.e., in case $\exists k \, \theta_k = 0$). For this paper, we will assume that all propensities are positive and thus $\hat{\Delta}_{IPS}$ is unbiased.

Finally, different methods have been proposed to optimize f based on $\hat{\Delta}_{IPS}$. Joachims et al. [13] show that Rank-SVM [11] can be adapted to optimize IPS estimates for the average-relevant-position metric. Agarwal et al. [1] introduce a method that can optimize any differentiable model w.r.t. an IPS estimate of a metric based on a monotonically decreasing function. Lastly, Oosterhuis and de Rijke [15] show that the supervised LambdaLoss LTR framework [22] can easily be adapted to optimize IPS estimates as well.

3 TRUST BIAS

Besides position bias, other forms of bias are also known to affect user interactions with ranked lists. Joachims et al. [12] conclude that the trust users have in a ranking system affects their click behavior. Because users trust the results, they are more likely to perceive top-ranked items to be relevant, even when the displayed information about the item suggests otherwise. Similar to position bias, this causes items displayed at high ranks to have an unfair advantage, however, despite this similarity the effects of the two types of bias are not identical.

Recently, Agarwal et al. [2] have modeled trust bias by distinguishing between *perceived relevance* $\tilde{R} \in \{0, 1\}$ and *real relevance* R. Trust bias occurs because users are more likely to perceive items as relevant $\tilde{R} = 1$ if they are among the top ranked items in the list. In Agarwal et al.'s model, a click happens when a user examines and perceives an item to be relevant: $C = 1 \leftrightarrow E = 1 \wedge \tilde{R} = 1$. The model combines rank-based position bias (as described in Section 2.2) with trust bias, resulting in the following click probability:

$$P(C = 1 \mid q, d, k) = P(E = 1 \mid k) \cdot P(\tilde{R} = 1 \mid E = 1, R, k). \tag{9}$$

Furthermore, the probability of perceived relevance of an examined item is conditioned on the actual relevance and the rank k at which item d is displayed. For brevity, we use ϵ_k^+ and ϵ_k^- to denote these probabilities:

$$\begin{aligned} P(\tilde{R} = 1 \mid E = 1, R = 1, k) &= \epsilon_k^+, \\ P(\tilde{R} = 1 \mid E = 1, R = 0, k) &= \epsilon_k^-. \end{aligned} \tag{10}$$

Additionally, we write $\gamma_{q,d}$ for the probability of actual relevance: $\gamma_{q,d} = P(R = 1 \mid q, d)$. These conventions allow us to have the following succinct notation for the click probability:

$$P(C = 1 \mid q, d, k) = \theta_k \left(\epsilon_k^+ \gamma_{q,d} + \epsilon_k^- (1 - \gamma_{q,d}) \right). \tag{11}$$

It is important to note that the combination of trust bias and position bias can be seen as an affine transformation between the relevance probabilities and click probabilities. If we choose $\alpha_k = \theta_k(\epsilon_k^+ - \epsilon_k^-)$ and $\beta_k = \theta_k \epsilon_k^-$, this affine transformation becomes apparent:

$$P(C = 1 \mid q, d, k) = \alpha_k P(R = 1 \mid q, d) + \beta_k. \tag{12}$$

We will use this property in Section 5 to introduce affine corrections for these biases.

An empirical analysis by Agarwal et al. [2] shows that their trust bias model better captures observed user behavior than the model that only considers position bias (Eq. 4). Furthermore, Agarwal et al. propose an IPS estimator in order to correct for both trust bias and position bias. In the next section, we will first prove that this estimator cannot correct for these biases. Moreover, we subsequently prove that no IPS estimator is capable of doing so. Then, in Section 5 we introduce an estimator based on affine corrections, and prove that it is the first unbiased estimator that corrects for both position bias and trust bias.

4 EXISTING METHODS AND TRUST BIAS

In this section, we discuss Agarwal et al. [2]'s Bayes-IPS method designed specifically for trust bias. We prove that no IPS estimator is able to correct for trust bias, including Bayes-IPS.

4.1 Bayes-IPS

Agarwal et al. [2] have proposed the Bayes-IPS estimator to correct for trust bias and position bias. This estimator combines two corrections: (i) correcting for position bias by weighting inversely to θ; and (ii) correcting for trust bias by weighting each click to the probability of true relevance: $P(R = 1 \mid \tilde{R} = 1, E = 1, k)$. This results in the following estimator:

$$\hat{\Delta}_{\text{Bayes-IPS}}(f) = \frac{1}{N} \sum_{i=1}^{N} \sum_{(d,k) \in y_i} \frac{\epsilon_k^+}{\epsilon_k^+ + \epsilon_k^-} \frac{c_i(d)}{\theta_k} \cdot \lambda(d \mid q_i, f). \tag{13}$$

We note that $\hat{\Delta}_{\text{Bayes-IPS}}$ is still an IPS estimator; the difference with $\hat{\Delta}_{IPS}$ is that it uses the weights $\frac{1}{\theta_k} \frac{\epsilon_k^+}{\epsilon_k^+ + \epsilon_k^-}$ instead of $\frac{1}{\theta_k}$. In addition to θ_k, Bayes-IPS also needs to know the values of ϵ_k^+ and ϵ_k^-. Agarwal et al. use Expectation Maximization (EM) to estimate these values from click logs, and, using the estimated values, optimize a ranking model using $\hat{\Delta}_{\text{Bayes-IPS}}$. Their results show that optimizing with $\hat{\Delta}_{\text{Bayes-IPS}}$ is more effective than using $\hat{\Delta}_{IPS}$ and leads to significant improvements when ranking for search through emails or other personal documents [2].

While empirical results indicate that $\hat{\Delta}_{Bayes\text{-}IPS}$ is an improvement over $\hat{\Delta}_{IPS}$, neither estimator is unbiased w.r.t. trust bias. If trust bias is present, i.e., if $\exists k, k' \ (\epsilon_k^- \neq \epsilon_{k'}^-)$, then $\hat{\Delta}_{Bayes\text{-}IPS}$ is biased. We can show this by looking at the difference between $\hat{\Delta}_{Bayes\text{-}IPS}$ and Δ, which is not necessarily equal to zero. Let $\lambda_{q,d}$ be short for $\lambda(d \mid q, f)$, then:

$$\begin{aligned}
&\Delta(f) - \mathbb{E}_{q,y,c}\left[\hat{\Delta}_{Bayes\text{-}IPS}(f)\right] \\
&= \mathbb{E}_{q,y,c}\left[\sum_{(d,k)\in y_i}\left(\gamma_{q,d} - \frac{\epsilon_k^+}{\epsilon_k^+ + \epsilon_k^-}\frac{P(C=1 \mid q,d,k)}{\theta_k}\right)\cdot\lambda_{q,d}\right] \quad (14)\\
&= \mathbb{E}_{q,y,c}\left[\sum_{(d,k)\in y_i}\left(\left(1 - \frac{\epsilon_k^+\left(\epsilon_k^+ - \epsilon_k^-\right)}{\epsilon_k^+ + \epsilon_k^-}\right)\gamma_{q,d} - \left(\frac{\epsilon_k^+\epsilon_k^-}{\epsilon_k^+ + \epsilon_k^-}\right)\right)\cdot\lambda_{q,d}\right].
\end{aligned}$$

Clearly, it is non-trivial to derive under what conditions the difference between $\Delta(f)$ and $\mathbb{E}_{q,y,c}[\hat{\Delta}_{Bayes\text{-}IPS}(f)]$ is zero. Instead of further investigating Bayes-IPS, we will prove that no IPS estimator is unbiased w.r.t. trust bias under non-trivial circumstances, thereby also proving that no practical conditions exist where this difference is guaranteed to be zero.

4.2 IPS cannot correct for trust bias

We proceed by considering whether any IPS estimator can be unbiased w.r.t. trust bias. Consider a generic IPS estimator $\hat{\Delta}_\rho$. We will derive the values the propensities ρ should have for unbiased CLTR:

$$\hat{\Delta}_\rho(f) = \frac{1}{N}\sum_{i=1}^{N}\sum_{(d,k)\in y_i}\frac{c_i(d)}{\rho_{q_i,d,k}}\cdot\lambda(d \mid q_i, f). \quad (15)$$

Importantly, we have to limit the possible choices for ρ, because trivially unbiased estimators are theoretically possible [9]:

$$\forall d,k \ \rho_{q,d,k} = \frac{\frac{1}{N}\sum_{i=1}^{N}\sum_{(d,k)\in y_i}\gamma_{d,k}\cdot\lambda(d \mid q_i, f)}{\frac{1}{N}\sum_{i=1}^{N}\sum_{(d,k)\in y_i}c_i(d)\cdot\lambda(d \mid q_i, f)}. \quad (16)$$

To avoid such trivial situations, we use the following definition for circumstances where CLTR is not a trivial problem:

Definition 1. We define *non-trivial circumstances* as situations where no information about the relevances γ is known. Furthermore, trust bias must be present, meaning users' trust must not be constant at all the ranks:

$$\exists k, k' \ (\epsilon_k^- \neq \epsilon_{k'}^-). \quad (17)$$

Additionally, every displayed item should have a chance of being clicked and clicks at any rank k should be positively correlated with relevance:

$$\forall k \ (\theta_k(\epsilon_k^+ - \epsilon_k^-) > 0). \quad (18)$$

Lastly, the metric λ should not be indifferent to the ranking of f:

$$\exists q, d, f, f' \ (\lambda(d \mid q, f) \neq \lambda(d \mid q, f')). \quad (19)$$

With this definition we avoid the following scenarios: (i) ρ is chosen based on the known values of γ, in which case there is no need to estimate $\Delta(f)$ based on clicks; (ii) there is no trust bias, in which case every method is trivially unbiased w.r.t. trust bias; (iii) some items cannot receive clicks or clicks are not indicative of relevance, in these cases there is no signal to learn from; (iv) the metric is indifferent to the ranking function f, in which case there is nothing to evaluate since all ranking functions are equally good.

Naturally, an unbiased estimator should lead to the same optimal ranking as the full information case. For this, it is sufficient to have consistent pairwise rankings. To be clear about what we are going to prove about unbiasedness w.r.t. trust bias, we introduce the following formal definition:

Definition 2. An IPS estimator $\hat{\Delta}_\rho$ is *unbiased w.r.t. trust bias*, if in all non-trivial circumstances ρ can be chosen so that it can correctly predict relative differences:

$$\exists\rho, \forall f, f', \ \left(\Delta(f) > \Delta(f') \leftrightarrow \mathbb{E}_c\left[\hat{\Delta}_\rho(f)\right] > \mathbb{E}_c\left[\hat{\Delta}_\rho(f')\right]\right). \quad (20)$$

In other words, we define an estimator to be unbiased w.r.t. to trust bias, if it can unbiasedly predict the preference between any two rankers under any non-trivial circumstances. Again, it is important to avoid ρ being chosen based on knowledge of γ. If a $\hat{\Delta}_\rho$ meets our definition of unbiasedness it can safely be applied in any non-trivial circumstances; we argue that this covers all realistic CLTR situations.

Theorem 1. *No IPS estimator is unbiased w.r.t. trust bias.*

Proof. We will prove this by showing that there are non-trivial circumstances where no values of ρ exist for $\hat{\Delta}_\rho$ to correctly predict relative differences. We do so by starting from the most basic ranking example and deriving the values of ρ where $\hat{\Delta}_\rho$ is unbiased, we prove that no such values exist. In addition to this proof, we will show how our basic example can be extended to include rankings with more queries and items.

In our basic example, we consider a system that only receives a single query q_1: $P(q_1) = 1$ and that only has to rank two documents $D_{q_1} = \{d_1, d_2\}$. Therefore, two ranking functions can cover all possible rankings: f_1 that produces $[d_1, d_2]$ and f_2 that produces $[d_2, d_1]$. Lastly, the metric we consider is a top-1 metric, which means it is only affected by the top document of a ranking:

$$\begin{aligned}
\lambda(d_1 \mid q_1, f_1) = \lambda(d_2 \mid q_1, f_2) > 0,\\
\lambda(d_2 \mid q_1, f_1) = \lambda(d_1 \mid q_1, f_2) = 0.
\end{aligned} \quad (21)$$

Thus, in this basic example, we are trying to estimate whether d_1 should be ranked higher than d_2 or vice-versa.

The true difference in metric value between the rankers is:

$$\Delta(f_1) - \Delta(f_2) = (\gamma_{q_1,d_1} - \gamma_{q_1,d_2})\cdot\lambda(d_1 \mid q_1, f_1). \quad (22)$$

Therefore, only the difference in item relevance matters for the relative difference:

$$\text{sign}\big(\Delta(f_1) - \Delta(f_2)\big) = \text{sign}(\gamma_{q_1,d_1} - \gamma_{q_1,d_2}). \quad (23)$$

The estimates of $\hat{\Delta}_\rho$ are based on N interactions with query q_1 where at each interaction d_1 and d_2 were displayed at rank 1 and 2, respectively. The difference in the expected estimates (cf. Eq. 11 and Eq. 15) is therefore:

$$\begin{aligned}
&\mathbb{E}_c\left[\hat{\Delta}_\rho(f_1)\right] - \mathbb{E}_c\left[\hat{\Delta}_\rho(f_2)\right]\\
&\quad= \frac{\theta_1\left(\left(\epsilon_1^+ - \epsilon_1^-\right)\gamma_{q_1,d_1} + \epsilon_1^-\right)}{\rho_{q_1,d_1,1}} - \frac{\theta_2\left(\left(\epsilon_2^+ - \epsilon_2^-\right)\gamma_{q_1,d_2} + \epsilon_2^-\right)}{\rho_{q_1,d_2,2}}.
\end{aligned} \quad (24)$$

We note that this scenario falls under the definition of a non-trivial circumstance (Definition 1).

In order to be unbiased, the values of the propensities ρ must be chosen so that the requirement in Eq. 20 is met. Note that for two continuous functions to always have the same sign, they should agree on zero values. By combining Eq. 20 with Eq. 23 and Eq. 24, we can derive that ρ must meet the following requirement:

$$\gamma_{q_1,d_1} = \gamma_{q_1,d_2} \leftrightarrow \frac{\rho_{q_1,d_1,1}}{\rho_{q_1,d_2,2}} = \frac{\theta_1\left(\left(\epsilon_1^+ - \epsilon_1^-\right)\gamma_{q_1,d_1} + \epsilon_1^-\right)}{\theta_2\left(\left(\epsilon_2^+ - \epsilon_2^-\right)\gamma_{q_1,d_2} + \epsilon_2^-\right)}. \tag{25}$$

Under non-trivial circumstances, ρ has to be chosen without knowledge of γ, therefore we must find a single value for each of $\rho_{q_1,d_1,1}$ and $\rho_{q_1,d_2,2}$ that meets this requirement for all possible values of γ. Combining this fact with the fact that γ consists of probabilities, we can derive the following requirement from Eq. 25:

$$\forall x \in [0,1]\left(\frac{\rho_{q_1,d_1,1}}{\rho_{q_1,d_2,2}} = \frac{\theta_1\left(\left(\epsilon_1^+ - \epsilon_1^-\right)x + \epsilon_1^-\right)}{\theta_2\left(\left(\epsilon_2^+ - \epsilon_2^-\right)x + \epsilon_2^-\right)}\right). \tag{26}$$

From this we can directly derive the following requirement for the bias parameters ϵ:

$$\forall x \in [0,1]\left(\frac{\epsilon_1^+}{\epsilon_2^+} = \frac{\epsilon_1^-}{\epsilon_2^-} = \frac{\left(\epsilon_1^+ - \epsilon_1^-\right)x + \epsilon_1^-}{\left(\epsilon_2^+ - \epsilon_2^-\right)x + \epsilon_2^-}\right). \tag{27}$$

Thus, a solution for the propensities ρ in Eq. 26 only exists if the trust bias parameters ϵ meet the requirement in Eq. 27. Solving for ϵ shows that the latter requirement can be simplified to:

$$\frac{\epsilon_1^+}{\epsilon_2^+} = \frac{\epsilon_1^-}{\epsilon_2^-}. \tag{28}$$

Therefore, only in very specific cases where trust bias adheres to Eq. 28 do values of ρ exist that can meet Eq. 25. This proves the theorem since we have provided input cases where no IPS is unbiased. In fact, in non-trivial circumstances (Definition 1 and Eq. 17), the probability of even being close to this regularity of Eq. 28 is so low that in practice we can safely say that it never happens. So, not only do non-trivial input cases exist where no IPS can be unbiased, but almost all of the time we are dealing with such cases.

Therefore, we have proven that $\hat{\Delta}_\rho$ can never correctly predict the relative difference between f_1 and f_2 in this example under non-trivial circumstances. In conclusion, we have therefore proven that no IPS estimator is unbiased w.r.t. trust bias, since there are examples where under non-trivial circumstances, no propensities ρ can be chosen so that it unbiasedly infers the preference between two rankers. □

While the basic counterexample used in the proof of Theorem 1 is enough for proving that IPS estimators are biased w.r.t. trust bias, we note that it can easily be extended to cases with more queries and items. For any number of queries and items and any item pair d_3 and d_4, there always exist two rankers f_3 and f_4 that agree on all item placements expect that they swap the ranks of d_3 and d_4. Using a proof analogous to the above, one can prove that similar to Eq. 28 $(\epsilon_{k_3}^+/\epsilon_{k_4}^+) = (\epsilon_{k_3}^-/\epsilon_{k_4}^-)$, where k_3 and k_4 are the display ranks of d_3 and d_4, respectively. This process can be repeated for other item pairs until the requirement $\forall k, k'\ ((\epsilon_k^+/\epsilon_{k'}^+) = (\epsilon_k^-/\epsilon_{k'}^-))$ is obtained. Thus, one can prove this very restrictive requirement to the trust bias, that applies regardless of the number of queries and documents. Only when the trust bias adheres to this requirement, is it possible that an IPS estimator may be able to correctly infer relative differences. This shows that IPS is not a practical solution to trust bias.

In summary, we have proven that no IPS estimator is unbiased w.r.t. trust bias without *a priori* knowledge of the relevance γ, and thus is not applicable in any practical circumstances. We have done so by taking a generic IPS estimator and deriving the possible values for the propensities ρ that would lead to unbiased results in the most basic ranking scenario. The proof of Theorem 1 shows that for most instances of trust bias such values do not exist. Thus, none of the existing IPS estimators can correct for trust bias or can be adapted to do so. For clarity, this includes: the original CLTR estimators [13, 20]; the dual learning algorithm by Ai et al. [4]; the IPS with corrections for item-selection bias by Ovaisi et al. [16]; the policy aware estimator [15]; and the Bayes-IPS estimator [2].

The problem with IPS appears to be that trust bias causes an affine transformation between relevance probabilities and click probabilities. For a single query item pair q, d displayed at rank k, ideally a propensity $\rho_{q,d,k}$ exists so that:

$$\gamma_{q,d} = \frac{\alpha_k\gamma_{q,d} + \beta_k}{\rho_{q,d,k}}. \tag{29}$$

Such a propensity does exist but it is dependent on γ:

$$\rho_{q,d,k} = \frac{\alpha_k\gamma_{q,d} + \beta_k}{\gamma_{q,d}}. \tag{30}$$

If $\beta_k = 0$ (i.e., $\epsilon_k^- = 0$), the transformation becomes linear and ρ becomes independent of γ: $\rho_{q,d,k} = \alpha_k$. Thus, the core issue is that IPS applies a linear transformation to observed clicks but a linear transformation cannot correct for the affine transformation caused by trust bias. As a solution to this problem, we will introduce a novel estimator that applies affine corrections to clicks.

5 AFFINE CORRECTIONS FOR TRUST BIAS

Next, we introduce our novel affine estimator: the first method that is proven to correct for trust bias. We also compare the affine estimator with the existing IPS estimator, and introduce an adaption of the EM algorithm for estimating trust bias.

5.1 The novel affine estimator

In Section 3 we described how trust bias can be seen as an affine transformation from relevance probabilities to click probabilities (see Eq. 12). Subsequently, in Section 4.2 we proved that IPS estimators cannot correct for trust bias because IPS can only apply linear transformations and no linear transformation can reverse the effect of an affine transformation (in non-trivial circumstances).

We now propose a novel estimator based on affine transformations to correct for both position bias and trust bias: the affine estimator. The estimator works for any situation where click probabilities are based on an affine transformation of relevance probabilities:

$$P\big(C = 1 \mid q, d, k\big) = \alpha_k P(R = 1 \mid q, d) + \beta_k. \tag{31}$$

This includes trust bias where $\alpha_k = \theta_k(\epsilon_k^+ - \epsilon_k^-)$ and $\beta_k = \theta_k\epsilon_k^-$. Spelled out in the notation of Eq. 31 and in the trust bias notation, the affine estimator is:

$$\begin{aligned}\hat{\Delta}_{\text{affine}}(f) &= \frac{1}{N}\sum_{i=1}^{N}\sum_{(d,k)\in y_i}\frac{c_i(d)-\beta_k}{\alpha_k}\cdot\lambda(d\mid q_i,f)\\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{(d,k)\in y_i}\frac{c_i(d)-\theta_k\epsilon_k^-}{\theta_k(\epsilon_k^+-\epsilon_k^-)}\cdot\lambda(d\mid q_i,f).\end{aligned} \tag{32}$$

We see that the affine estimator reweights clicks inversely to α_k, which is somewhat similar to IPS. However, the salient difference is that $\hat{\Delta}_{\text{affine}}$ also penalizes items by subtracting $\frac{\beta_k}{\alpha_k}$. This penalty compensates for *incorrect* clicks where the perceived relevance does not match the true relevance: $\tilde{R} = 1 \wedge R = 0$. Thus items displayed at ranks where more *incorrect* clicks take place receive more penalties, while simultaneously clicks are reweighted according to the position bias θ_k and to compensate for the penalties: $\epsilon_k^+ - \epsilon_k^-$. We note that unlike with the IPS estimator, an item that is displayed but not clicked can receive a negative weight. In expectation later clicks will compensate for this effect.

Theorem 2. *The affine estimator is unbiased w.r.t trust bias.*

Proof. First, we use the assumption that clicks are correlated with relevancy: $\forall k\ (\alpha_k \neq 0)$. Then we consider the expected value for a single click $c_i(d)$:

$$\mathbb{E}_c\left[\frac{c_i(d)-\beta_k}{\alpha_k}\right] = \frac{(\alpha_k\cdot\gamma_{q,d}+\beta_k)-\beta_k}{\alpha_k} = \gamma_{q,d}. \tag{33}$$

We can use this to derive the expected value of the affine estimator; it is equal to the true metric value:

$$\begin{aligned}&\mathbb{E}_{q,y,c}\left[\hat{\Delta}_{\text{affine}}(f)\right]\\ &\quad= \mathbb{E}_{q,y}\left[\sum_{(d,k)\in y_i}\mathbb{E}_c\left[\frac{c_i(d)-\beta_k}{\alpha_k}\right]\cdot\lambda(d\mid q,f)\right]\\ &\quad= \mathbb{E}_{q,y}\left[\sum_{(d,k)\in y_i}\gamma_{q,d}\cdot\lambda(d\mid q,f)\right] = \Delta(f).\end{aligned} \tag{34}$$

Therefore, the affine estimator is unbiased in expectation. □

The negative penalties (β_k/α_k) may be counter-intuitive. For a better understanding we consider a maximally non-relevant item $\gamma_{q,d} = 0$ that is displayed at rank k, M times. We expect to observe $M\cdot\beta_k$ clicks (all incorrect since $\gamma_{q,d} = 0$). The sum of the penalties for the item given by the affine estimator is $M\cdot(\beta_k/\alpha_k)$, while each click is weighted by $1/\alpha_k$. Thus, if we sum the weights for the clicks we expect $M\cdot(\beta_k/\alpha_k)$, therefore taking this sum minus the penalties correctly results in a zero weight for the item (in expectation). As with any estimator, for reliable estimates, M needs to be considerably large due to variance.

This concludes the introduction of our novel affine estimator. By performing affine transformations to clicks it is the first estimator that can correct for the effect of both position bias and trust bias.

5.2 Relation with IPS and other properties

While the affine estimator is very distinct from IPS since it can perform corrections that IPS cannot, we consider the former to be an extension of the latter. In the most straightforward way any IPS-based estimator can be seen as a special case of the affine estimator where $\forall k\ (\beta_k = 0)$. More generally, we consider the situation without trust bias, i.e., where ϵ_k^+ and ϵ_k^- have the same value for every k: $\forall k, k'\ (\epsilon_k^- = \epsilon_{k'}^- = \epsilon^- \wedge \epsilon_k^+ = \epsilon_{k'}^+ = \epsilon^+)$ and where clicks are positively correlated with relevance: $\epsilon^+ > \epsilon^-$. Also, we will assume that summing λ over documents leads to a constant value:

$$\forall f, f', q\left(\sum_{d\in D_q}\lambda(d\mid q,f) = \sum_{d\in D_q}\lambda(d\mid q,f')\right). \tag{35}$$

This means that if all items are equally relevant the order of the items does not matter. We note that this holds for virtually all ranking metrics, e.g., DCG, precision, recall, MAP, ARP, etc. Now, Eq. 32 can be rewritten as follows:

$$\begin{aligned}&\hat{\Delta}_{\text{affine}}(f)\\ &\quad= \frac{1}{N}\frac{1}{\epsilon^+-\epsilon^-}\sum_{i=1}^{N}\sum_{(d,k)\in y_i}\left(\frac{c_i(d)}{\theta_k}-\epsilon^-\right)\cdot\lambda(d\mid q_i,f)\\ &\quad= \frac{1}{\epsilon^+-\epsilon^-}\hat{\Delta}_{IPS}(f) - \frac{1}{N}\frac{\epsilon^-}{\epsilon^+-\epsilon^-}\sum_{i=1}^{N}\sum_{(d,k)\in y_i}\lambda(d\mid q_i,f)\\ &\quad= \frac{1}{\epsilon^+-\epsilon^-}\hat{\Delta}_{IPS}(f) - \mathbb{C},\end{aligned} \tag{36}$$

where $\mathbb{C}$ is a constant independent of f. Therefore, $\hat{\Delta}_{\text{affine}}$ unbiasedly predicts relative differences w.r.t. $\hat{\Delta}_{IPS}$:

$$\forall f, f',\ \hat{\Delta}_{\text{affine}}(f) > \hat{\Delta}_{\text{affine}}(f') \leftrightarrow \hat{\Delta}_{IPS}(f) > \hat{\Delta}_{IPS}(f'). \tag{37}$$

Consequently, we can conclude that optimizing f w.r.t. $\hat{\Delta}_{\text{affine}}(f)$ also optimizes w.r.t. $\hat{\Delta}_{IPS}$ when trust bias is not present. This further shows that the affine estimator should be viewed as a generalization of the existing IPS approach.

Furthermore, the notation of the affine estimator in Eq. 32 also reveals some other intuitive properties. We see that if for some k, $\alpha_k = 0$, then the estimator becomes undefined, thus if clicks are not correlated with relevance, the estimator cannot be applied. Interestingly, if we compare this with the trust bias model we see that there are only two cases when $\exists k\ (\alpha_k = 0)$ can occur: (i) when $\exists k\ (\theta_k = 0)$, i.e., at some rank k some items cannot be observed or clicked, hence nothing about the item at this rank can be learned; or (ii) when $\exists k\ (\epsilon_k^+ = \epsilon_k^-)$, i.e., at some rank k non-relevant and relevant items are equally likely to be clicked, thus there is nothing to learn from the click signal. Furthermore, something interesting happens if $\exists k\ (\epsilon_k^+ < \epsilon_k^-)$, i.e., if at some rank k *non-relevant* items are *more* likely to be clicked than *relevant* items. In this case, non-clicked items receive a positive penalty and clicks lead to negative scores, meaning the less clicked items are preferred since they are more likely to be relevant. All these cases are very intuitive and we consider it a great strength that they can be inferred from the affine estimator from just a brief analysis of its formulation.

5.3 Parameter estimation

Agarwal et al. [2] describe how EM can be used to estimate the position bias and trust bias parameters. We also use the regression-based EM procedure for estimating the bias parameters. However, unlike Agarwal et al., who estimate three parameters per rank k, namely θ_k, ϵ_k^- and ϵ_k^+, we notice that only two have to be estimated:

$$\begin{aligned}\zeta_k^+ &= P(C=1 \mid R=1,k) = \theta_k \epsilon_k^+ = \alpha_k + \beta_k \\ \zeta_k^- &= P(C=1 \mid R=0,k) = \theta_k \epsilon_k^- = \beta_k.\end{aligned} \tag{38}$$

From these two parameters the value of α_k and β_k can be inferred directly ($\alpha_k = \zeta_k^+ - \zeta_k^-$), and these are the only parameters required for the trust bias click model (Eq. 12) and the affine estimator (Eq. 32).

To estimate these parameters we adapt the Expectation step, where the parameters are updated as follows:

$$\zeta_k^+ = \frac{\sum_{i=1}^N c_i(d) P(R=1|C=1,q_i,d,k)}{\sum_{i=1}^N c_i(d) P(R=1|C=1,...) + (1-c_i(d)) P(R=1|C=0,...)}, \tag{39}$$

and

$$\zeta_k^- = \frac{\sum_{i=1}^N c_i(d) P(R=0 \mid C=1,q_i,d,k)}{\sum_{i=1}^N c_i(d) P(R=0|C=1,...) + (1-c_i(d)) P(R=0|C=0,...)}, \tag{40}$$

where the conditional relevance probabilities $P(R \mid C,q,d,k)$ are computed using Bayes's law. This simplification allows us to estimate the parameters with less computational costs. And since fewer parameters are estimated, we expect EM to converge faster.

In the Maximization step, the γ values are estimated by a regression algorithm. We use ζ^- and ζ^+ obtained from the E-step to train the unbiased ranking function f based on $\hat{\Delta}_{\text{affine}}$. Previous work [2, 21] suggests to use a *sigmoid* as a final activation function to obtain valid probability values. However, we observed that the sigmoid function gives very similar relevance probabilities between most items. In contrast, the *softmax* function results in more varied values but it forces the probabilities to sum to one for each query. As a simple alternative we propose the *soft-min-max* function, which does not force probabilities to sum to one, but still results in varied values:

$$\text{soft-min-max}(x_i) = \frac{e^{x_i} - e^{\min(x_i)}}{e^{\max(x_i)} - e^{\min(x_i)}}. \tag{41}$$

Our experiments show that the choice of activation function leads to noticeable differences.

6 EXPERIMENTAL SETUP

We follow the semi-synthetic setup that is prevalent in existing CLTR work [4, 10, 13, 15], where queries, documents and relevances are sampled from supervised LTR datasets, while clicks are simulated using probabilistic user models. First, we train a production ranker for each dataset; we randomly select 20 queries from each training set and use the supervised LTR LambdaMART method to optimize a ranking model. With these production rankers we simulate a situation where a decent ranking system exists but still leaves plenty of room for improvement. On each dataset, we simulate user interactions by repeatedly: (i) uniform-random sampling a query from the training set, (ii) ranking the documents for that query with the production ranker, and (iii) simulating clicks on the resulting ranking using a probabilistic user model. This semi-synthetic setup allows us to vary the number of clicks available for learning, as well as the position bias and trust bias of the simulated user. Thus, we can analyze the effects these factors have on the affine estimator and other CLTR methods.

6.1 Datasets

We use two of the largest publicly available LTR datasets: Yahoo! Webscope [5] and MSLR-WEB30k [17]. Both were created by a commercial search engine, and each contains around 30 000 queries, each query has a set of preselected documents to be ranked. The datasets contain five level relevancy tags acquired through expert labelling for the preselected query-document pairs. Yahoo! has 24 documents per query on average and uses 700-feature vectors to represent query-documents; MSLR has 125 per query and uses 136 features. Each dataset is split in training, validation and test sets; we only use the first fold of MSLR.

6.2 Click simulation

Clicks are simulated on rankings produced by the production rankers by applying probabilistic click models.

Per experimental setting, we simulate up to $8 \cdot 10^6$ clicks on the training set. The number of validation clicks is always 15% and 33% of the training clicks for Yahoo! and MSLR, respectively. These numbers were chosen to match the ratio between the number of training and validation queries in each dataset.

We apply Agarwal et al. [2]'s trust bias model with varying parameters (see Section 3). The relevances $\gamma_{q,d}$ are based on the relevance label recorded in the datasets; we follow Joachims et al. [13] and use binary relevance:

$$P(R=1 \mid q,d) = \gamma_{q,d} = \begin{cases} 1 & \text{if relevance\_label}(q,d) > 2, \\ 0 & \text{otherwise.} \end{cases} \tag{42}$$

Similar to previous work [10, 13, 15], we set the position bias inversely proportional to the display rank:

$$P(E=1 \mid k) = \theta_k = \left(\frac{1}{\min(k,20)}\right)^\eta, \tag{43}$$

where we vary the η parameter: $\eta \in \{1,2\}$.

To the best of our knowledge, this is the first CLTR that simulates trust bias, thus there is no precedent for the values of ϵ_k^+ and ϵ_k^-. In order to simulate trust bias as realistically as possible, we base our values on the empirical work of Agarwal et al. [2]. It appears that the bias Agarwal et al. inferred from actual user interactions can be approximated by the following formula:

$$\forall k \in \{1,2,\ldots,5\}, \quad \epsilon_k^+ \approx 1 - \frac{k+1}{100} \quad \wedge \quad \epsilon_k^- \approx \epsilon_1^- \frac{1}{k}. \tag{44}$$

Unfortunately, Agarwal et al. only observed interactions on top-5 rankings. To prevent ϵ_k^+ and ϵ_k^- from disappearing on ranks beyond $k=5$, we apply the following

$$\epsilon_k^+ = 1 - \frac{\min(k,20)+1}{100}, \quad \epsilon_k^- = \epsilon_1^- \frac{1}{\min(k,10)}. \tag{45}$$

We use the *incorrect-click* rate on the first rank: ϵ_1^-, as a hyper-parameter to vary the amount of trust bias. We found that our results are consistent across different values for ϵ_1^-. To cover both

cases with high and low trust bias, we report results with $\epsilon_1^- \in \{0.65, 0.35\}$.

6.3 LTR algorithm

Similar to Ai et al. [4] and Agarwal et al. [1] we train neural networks for our ranking functions. Our preliminary results indicate that the configuration of the networks does not have to be fine-tuned. The reported results are produced using models with three hidden layers with sizes $[512, 256, 128]$ respectively. All layers use *elu* activations and 0.1 dropout was applied to the last two layers.

For the loss function we follow Oosterhuis and de Rijke [15] and use LambdaLoss to optimize DCG [22]. For updating the gradients, we use the AdaGrad optimizer [7] with a learning rate of 0.004 and 0.02 for Yahoo! and MSLR datasets respectively, for 32 epochs.

6.4 Experimental runs

We evaluate the performance of our affine estimator, by comparing the nDCG@10 of the models it produces with those produced using other estimators. The following estimators are used as baselines:

(1) **No Correction**: The naïve estimator where each click is treated as an unbiased relevance signal.
(2) **IPS**: The original CLTR IPS estimator [13, 21] that only corrects for position bias (see Section 2.2).
(3) **Bayes-IPS**: The only existing CLTR estimator [2] designed for addressing trust bias (see Section 4).

For a clearer analysis, we also report the performance of the following ranking models:

(4) **Production**: The production ranker used in during the logging of simulated clicks.
(5) **Full Info**: A model trained using supervised LTR on the true relevance probabilities, its performance illustrates the (theoretical) maximal performance possible on a dataset. We note that this is not a baseline as it does not learn from clicks but (unrealistically) from the true relevances.

All reported nDCG@10 results are an average of four independent runs. Our experiments cover both the situation where the bias (θ, ϵ^- and ϵ^+) is known, e.g., through previous experiments [3, 8, 21], and the situation where the bias has to be estimated still.

7 RESULTS AND DISCUSSION

This section discusses our experimental results. We consider the ranking performance of the affine estimator compared to other estimators, in both the situation where the exact bias is known and where it has to be estimated.

7.1 Optimization with the affine estimator

First we consider whether *optimizing with the affine estimator leads to better performing ranking models than with existing estimators.*

Figure 1 shows the performance (nDCG@10) reached by the different estimators under varying degrees of bias and different numbers of clicks available for training. We see that the naïve estimator has already converged after $3 \cdot 10^5$ clicks, since additional clicks do not increase its performance. In line with the empirical results of Agarwal et al. [2], we see that both IPS and Bayes-IPS improve over the naïve estimator, and that Bayes-IPS consistently outperforms IPS. However, when we compare with the Full Info ranker, we see that there is still a sizable gap between Full Info and Bayes-IPS in every tested setting on both datasets. In other words, neither IPS nor Bayes-IPS can approximate the optimal model under the tested degrees of trust bias. As predicted by the theory in Section 4, it thus appears that both these IPS estimators are biased w.r.t. trust bias.

In contrast, we see that the affine estimator does approximate the optimal model when position bias is mild ($\eta = 1$). However, under extreme position bias ($\eta = 2$) it has not reached convergence in any of our graphs. Based on the theory in Section 5.1, we expect convergence near the optimal model if it were given more training clicks. Furthermore, in all tested settings we observe the affine estimator to outperform the other estimators when more than 10^6 training clicks are available. Using the Student's t-test we found that all the improvements at $8 \cdot 10^6$ clicks are significant with $p \leq 0.001$, except for the results on MSLR-WEB30k with $\eta = 1$ and $\epsilon_1^- = 0.35$ with a significance of $p \leq 0.002$. On small numbers of training clicks, the affine estimator has a similar or slightly lower performance than the other estimators. This could be explained by the bias-variance tradeoff: the Bayes-IPS and IPS estimators could have lower variance due to their bias, making them perform better on small amounts of data. Potentially, using propensity clipping on the affine estimator can increase its performance here [19].

In conclusion, our results strongly indicate that optimizing with the affine estimator results in better performing ranking models than with previously proposed estimators. In particular, on both datasets we see that, given enough click data, the affine estimator can be used to approximate the optimal ranking model, in settings with high or low degrees of trust bias or position bias.

7.2 Optimization with estimated biases

Next, we consider whether *optimization with the affine estimator is robust to estimated bias values.* This is important as in practice the values of bias parameters have to be estimated as well. While the theory proves that the affine estimator is unbiased when provided with the true bias values, we will now investigate whether it is still effective when they are estimated.

Figure 2 shows the performance (nDCG@10) reached by the affine estimator using bias parameters estimated from clicks (see Section 5.3), under varying degrees of position and trust bias. For clarity, both the ranking model optimization and the bias parameter estimation used the same clicks. Furthermore, the results in Figure 2 are separated for different final activation functions. Figure 3 shows the estimated parameters after $8 \cdot 10^6$ clicks in the same settings.

In Figure 2 we see that parameter estimation with the soft-min-max function leads to the best performance: soft-min-max outperforms the other functions in all settings, regardless of the number of training clicks. Though the difference between soft-min-max and optimization with the true bias values is noticeable, it appears to be a small difference, especially after 10^6 clicks. This suggests that the affine estimator with the soft-min-max function is robust to estimated bias values. Additionally, we see that the softmax function leads to decent performance when many clicks are available, but handles small numbers of clicks less well. Lastly, the sigmoid function results in the poorest performance.

Interestingly, Figure 3 shows that none of the functions leads to extremely accurate bias estimation with EM. We see that except for the first position, Soft-min-max and sigmoid underestimate

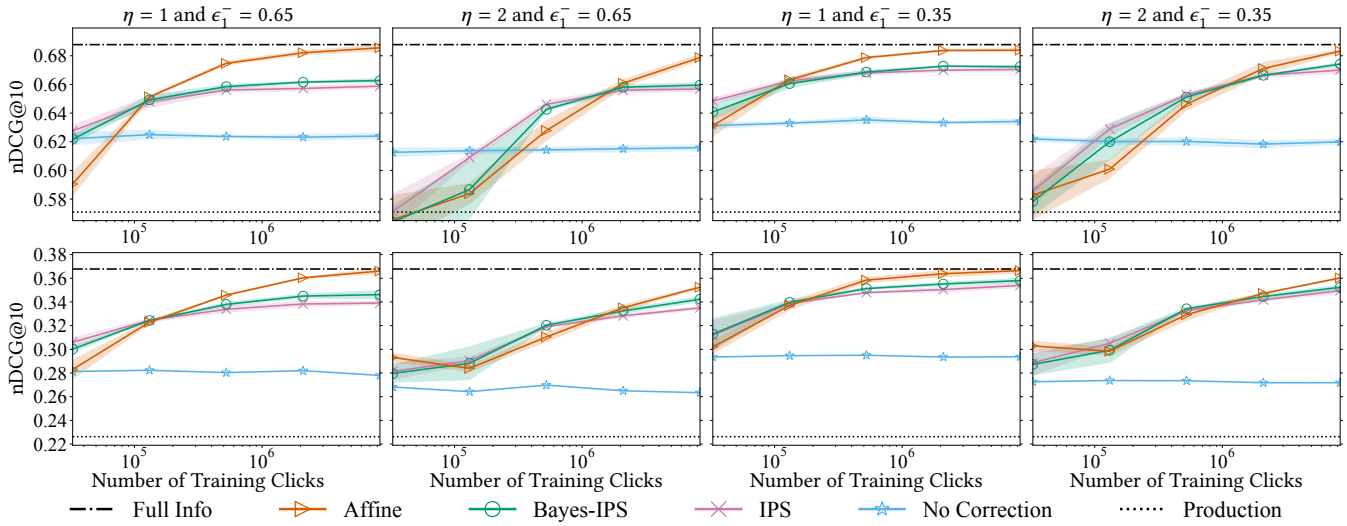

Figure 1: Comparison of different CLTR estimators in term of nDCG@10 on different numbers of clicks and under varying levels of position bias and trust bias. Estimators were given the true bias parameters. Results are averaged over four runs; shaded area indicates the standard deviation. Top row: Yahoo! Webscope dataset; bottom row: MSLR-WEB30k dataset.

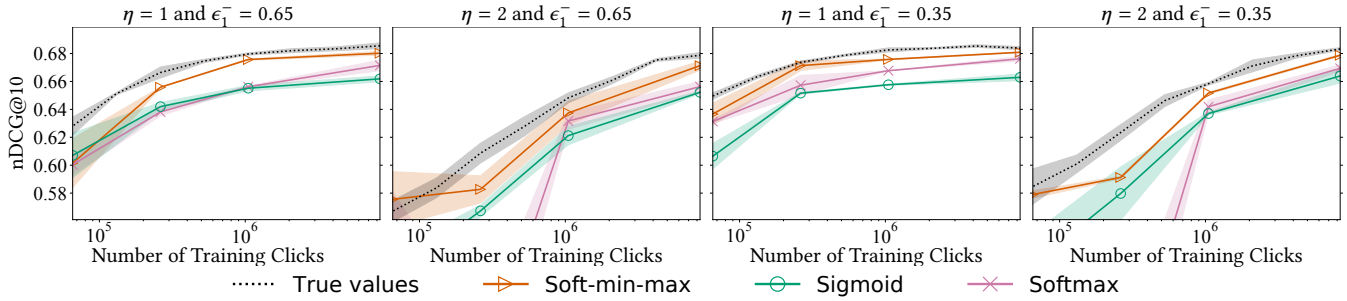

Figure 2: Comparison of different final activation functions to estimate the bias parameters, under varying levels of position and trust bias. Y-axis indicates the performance of ranking models optimized using the affine estimator. Results are averaged over four runs; shaded area indicates the standard deviation. All results are based on the Yahoo! Webscope dataset.

the values of α_k, while softmax overestimates it. While soft-min-max and softmax have accurate estimates of β_k, sigmoid appears to underestimate it. This further shows that the affine estimator is robust to estimated values, since soft-min-max leads to good performance while underestimating α_k. This seems to suggest that having an accurate estimate of β_k is more important than one for α_k. In theory, from Eq. 32 we see that, unlike β_k, α_k can be estimated within a constant factor of the true value without hurting the performance of $\hat{\Delta}_{\text{affine}}$. However, further analysis is required to fully understand what kind of inaccuracies still result in high performance. These results also suggest that there are promising opportunities for novel ways to estimate trust bias from click data.

In conclusion, our results show that using the affine estimator still leads to good performance when it is based on estimated bias values. In particular, we have found that using the soft-min-max function leads to the best results, and that the affine estimator can still get near-optimal performance when bias values are not completely accurate. We conclude that the affine estimator is robust w.r.t. estimated bias values.

8 CONCLUSION

In this paper we have considered CLTR in situations with both position bias and trust bias. We have proven that no IPS estimator can correct for trust bias, including the Bayes-IPS estimator specifically designed for this bias [2]. The reason for this inability is that trust bias is an affine transformation between relevance probabilities and click probabilities, and IPS estimators can only correct for linear transformations.

As a solution, we have introduced the novel affine estimator, which applies affine transformations to clicks: it both reweights clicks and penalizes items for being displayed at ranks where the users' trust is high. We proved that the affine estimator is unbiased w.r.t. both position bias and trust bias, thus it is the first CLTR method that can deal with both of these biases simultaneously. Furthermore, the affine estimator can be considered an extension of the existing IPS approach: when no trust bias is present the affine estimator optimizes the same objective as the existing IPS estimator. Our experimental results show that using the affine estimator CLTR can approximate the optimal model when both position bias and trust bias are present, while existing IPS-based estimators cannot.

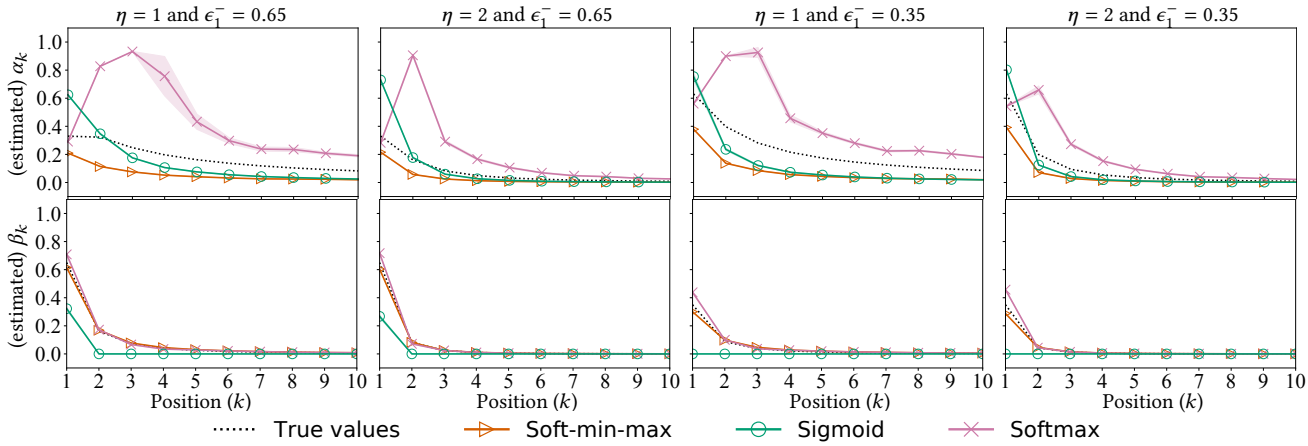

Figure 3: Bias parameters estimated on $8 \cdot 10^6$ clicks using different final activation functions, under varying degrees of position bias and trust bias. Results are averaged over four runs; shaded area indicates the standard deviation. All results are based on the Yahoo! Webscope dataset. Top row: α_k; bottom row: β_k.

Furthermore, our results suggest that the estimator is robust to bias estimation, as performance is stable when the bias parameters are estimated from interactions.

With the introduction of our affine estimator, the CLTR framework has been expanded to correct for trust bias on top of position bias. Future work can continue this trend, for instance, by combining the policy-aware approach by Oosterhuis and de Rijke [15] with the affine estimator, perhaps an estimator that corrects for both item-selection bias and trust bias can be found. Furthermore, previous work has found position bias estimation using randomization to be very powerful [3, 21]. Thus, there seems to be potential for methods based on randomization for estimating trust bias, possibly another fruitful direction for future research.

CODE AND DATA

To facilitate the reproducibility of the reported results, this work only made use of publicly available data and our experimental implementation is publicly available at https://github.com/AliVard/trust-bias-CIKM2020.

ACKNOWLEDGMENTS

This research was supported by Elsevier and the Netherlands Organisation for Scientific Research (NWO) under project nrs 652.002.001 and 612.001.551. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Aman Agarwal, Kenta Takatsu, Ivan Zaitsev, and Thorsten Joachims. 2019. A General Framework for Counterfactual Learning-to-Rank. In *SIGIR*. ACM, 5–14.
[2] Aman Agarwal, Xuanhui Wang, Cheng Li, Michael Bendersky, and Marc Najork. 2019. Addressing Trust Bias for Unbiased Learning-to-Rank. In *The World Wide Web Conference*. ACM, 4–14.
[3] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. 2019. Estimating Position Bias without Intrusive Interventions. In *WSDM*. ACM, 474–482.
[4] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased Learning to Rank with Unbiased Propensity Estimation. In *SIGIR*. ACM, 385–394.
[5] Olivier Chapelle and Yi Chang. 2011. Yahoo! Learning to Rank Challenge Overview. *Journal of Machine Learning Research* 14 (2011), 1–24.
[6] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An Experimental Comparison of Click Position-bias Models. In *WSDM*. 87–94.
[7] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research* 12, Jul (2011), 2121–2159.
[8] Zhichong Fang, Aman Agarwal, and Thorsten Joachims. 2019. Intervention Harvesting for Context-dependent Examination-bias Estimation. In *SIGIR*. 825–834.
[9] 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*. ACM, 2830–2836.
[10] 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 *SIGIR*. ACM, 15–24.
[11] Thorsten Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. In *KDD*. ACM, 133–142.
[12] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately Interpreting Clickthrough Data as Implicit Feedback. In *SIGIR*. ACM, 154–161.
[13] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *WSDM*. ACM, 781–789.
[14] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval* 3, 3 (2009), 225–331.
[15] Harrie Oosterhuis and Maarten de Rijke. 2020. Policy-Aware Unbiased Learning to Rank for Top-k Rankings. In *SIGIR*. ACM.
[16] Zohreh Ovaisi, Ragib Ahsan, Yifan Zhang, Kathryn Vasilaky, and Elena Zheleva. 2020. Correcting for Selection Bias in Learning-to-rank Systems. *arXiv preprint arXiv:2001.11358* (2020).
[17] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 datasets. *arXiv preprint arXiv:1306.2597* (2013).
[18] Mark Sanderson. 2010. Test Collection Based Evaluation of Information Retrieval Systems. *Foundations and Trends in Information Retrieval* 4, 4 (2010), 247–375.
[19] Adith Swaminathan and Thorsten Joachims. 2015. Batch Learning from Logged Bandit Feedback through Counterfactual Risk Minimization. *Journal of Machine Learning Research* 16, 1 (2015), 1731–1755.
[20] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to Rank with Selection Bias in Personal Search. In *SIGIR*. ACM, 115–124.
[21] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position Bias Estimation for Unbiased Learning to Rank in Personal Search. In *WSDM*. ACM, 610–618.
[22] Xuanhui Wang, Cheng Li, Nadav Golbandi, Michael Bendersky, and Marc Najork. 2018. The LambdaLoss Framework for Ranking Metric Optimization. In *CIKM*. ACM, 1313–1322.
[23] Yisong Yue, Rajan Patel, and Hein Roehrig. 2010. Beyond Position Bias: Examining Result Attractiveness as a Source of Presentation Bias in Clickthrough Data. In *WWW*. ACM, 1011–1018.