Contrasting Neural Click Models and Pointwise IPS Rankers

Philipp Hager[1 [0000−0001−5696−9732]], Maarten de Rijke[1 [0000−0002−1086−0202]], and Onno Zoeter[2 [0000−0003−1704−706X]]

[1] University of Amsterdam, The Netherlands
{p.k.hager,m.derijke}@uva.nl
[2] Booking.com, The Netherlands
onno.zoeter@booking.com

Abstract. Inverse-propensity scoring and neural click models are two popular methods for learning rankers from user clicks that are affected by position bias. Despite their prevalence, the two methodologies are rarely directly compared on equal footing. In this work, we focus on the pointwise learning setting to compare the theoretical differences of both approaches and present a thorough empirical comparison on the prevalent semi-synthetic evaluation setup in unbiased learning-to-rank. We show theoretically that neural click models, similarly to IPS rankers, optimize for the true document relevance when the position bias is known. However, our work also finds small but significant empirical differences between both approaches indicating that neural click models might be affected by position bias when learning from shared, sometimes conflicting, features instead of treating each document separately.

1 Introduction

Learning-to-rank a set of items based on their features is a crucial part of many real-world search [9, 23, 37, 42] and recommender systems [15, 20, 55]. Traditional supervised learning-to-rank uses human expert annotations to learn the optimal order of items [8, 9, 31]. However, expert annotations are expensive to collect [9] and can be misaligned with actual user preference [41]. Instead, the field of unbiased learning-to-rank seeks to optimize ranking models from implicit user feedback, such as clicks [1, 28, 34, 49, 50]. One well-known problem when learning from click data is that the position at which an item is displayed affects how likely a user is to see and interact with it [16, 27, 28, 47, 50]. Click modeling [14, 16, 19, 21, 39] and inverse-propensity scoring (IPS) [1, 25, 28, 35, 45] are two popular methods for learning rankers from position-biased user feedback. IPS-based counterfactual learning-to-rank methods mitigate position bias by re-weighting clicks during training inversely to the probability of a user observing the clicked item [28, 49]. In contrast, click models are generative models that represent position bias and item relevance as latent parameters to directly predict biased user behavior [14, 16, 19, 21, 39].

IPS approaches were introduced to improve over click models [28, 49] by: (i) requiring less observations of the same query-document pair by representing

items using features instead of inferring a separate relevance parameter for each document [1, 28, 49, 50]; (ii) decoupling bias and relevance estimations into separate steps since the joint parameter inference in click models can fail [2, 32, 50]; and (iii) optimizing the order of documents through pairwise [25, 28] and listwise loss [34] functions instead of inferring independent pointwise relevance estimations for each document [28, 50].

At the same time, neural successors of click models have been introduced [7, 13, 22, 23, 54, 55] that can leverage feature inputs, similarly to IPS-based rankers. Moreover, pointwise IPS methods have been presented that address the same ranking setting as click models [5, 40]. In this work, we ask if both approaches are two sides of the same coin when it comes to pointwise learning-to-rank?

To address this question, we first introduce both approaches (Sections 2, 3) and show theoretically that both methods are equivalent when the position bias is known (Section 4). We then compare both approaches empirically on the prevalent semi-synthetic benchmarking setup in unbiased learning-to-rank (Section 5) and find small but significant differences in ranking performance (Section 6.1). We conclude by investigating the found differences by performing additional experiments (Section 6.2) and hypothesize that neural click models might be affected by position bias when learning from shared, sometimes conflicting, document features.

The main contributions of this work are:

(1) A theoretical analysis showing that a PBM click model optimizes for unbiased document relevance when the position bias is known.
(2) An empirical evaluation of both methods on three large semi-synthetic click datasets revealing small but significant differences in ranking performance.
(3) An analysis of the empirical differences that hint at neural click models being affected by position bias when generalizing over conflicting document features instead of treating each document separately.

2 Related work

We provide an overview of probabilistic and neural click models, IPS-based counterfactual learning-to-rank, and comparisons between the two methodologies.

Click models. Probabilistic click models emerged for predicting user interactions in web search [14, 16, 39]. Factors that impact a user's click decision, such as an item's probability to be seen or its relevance are explicitly modeled as random variables, which are jointly inferred using maximum likelihood estimation on large click logs [14]. An early but prevailing model is the position-based model (PBM), which assumes that a click on a given item only depends on its position and relevance [16, 39]. Another prominent approach, the cascade model, assumes that users scan items from top to bottom and click on the first relevant item, not examining the documents below [16]. Follow-up work extends these approaches to more complex click behavior [11, 19, 21, 48], more elaborate user interfaces [52, 53], and feedback beyond clicks [18]. We refer to Chuklin et al. [14] for an overview.

Recent click models use complex neural architectures to model non-sequential browsing behavior [56] and user preference across sessions [12, 30]. Additionally, exact identifiers of items are typically replaced by more expressive feature representations [22, 54, 56]. In contrast to ever more complicated click models, neural implementations of the classic PBM recently gained popularity in industry applications [23, 54, 55]. So-called two-tower models input bias and relevance-related features into two separate networks and combine the output to predict user clicks [22, 54]. We use a neural PBM implementation similar to current two-tower models in this work and our findings on click model bias might be relevant to this community.

Counterfactual learning-to-rank. Joachims et al. introduced the concept of counterfactual learning-to-rank [28], relating to previous work by Wang et al. [49]. This line of work assumes a probabilistic model of user behavior, usually the PBM [25, 28, 34, 40] or cascade click model [46], and uses inverse-propensity scoring to mitigate the estimated bias from click data. The first work by Joachims et al. [28] introduced an unbiased version of the pairwise RankSVM method, Hu et al. [25] introduced a modified pairwise LambdaMART, and Oosterhuis and de Rijke suggested an IPS-correction for the listwise LambdaLoss framework [35]. Given that click models are pointwise rankers [50], we use a pointwise IPS method introduced by Bekker et al. [5] and later Saito et al. [40].

Comparing click models and IPS. Lastly, we discuss related work comparing IPS and click models. To our knowledge, Wang et al. [50] conduct the only experiment that compares both approaches on a single proprietary dataset. Their RegressionEM approach extends a probabilistic PBM using logistic regression to predict document relevance from item features instead of inferring separate relevance parameters per document. While the main motivation behind their work is to obtain better position bias estimates to train a pairwise IPS model, the authors also report the ranking performance of the inferred logistic regression model which can be seen as a component of a single-layer neural click model. The authors find that the click model improves rankings over a baseline not correcting for position bias, but is outperformed by a pairwise IPS approach [50, Table 4]. The authors also include two pointwise IPS approximations which are less effective than the click model and also fail to outperform the biased baseline model. Therefore, it is unclear how current pointwise methods suggested by Bekker et al. [5] and Saito et al. [40] would compare. We compare a recent pointwise IPS method with a common neural PBM implementation and report experiments on three public LTR dataset unifying model architecture, hyperparameter tuning, and position bias estimation to avoid confounding factors.

Lastly, recent theoretical work by Oosterhuis [32] compares click models and IPS and their limits for unbiased learning-to-rank. Their work finds that IPS-based methods can only correct for biases that are an affine transformation of item relevance. For click models jointly inferring both relevance and bias parameters, they find no robust theoretical guarantees of unbiasedness and find settings in which even an infinite amount of clicks will not lead to inferring the true model parameters. We will discuss this work in more detail in Section 4 and

extend their analysis to show that a click model only inferring item relevance should be in-fact unbiased.

3 Background

We introduce our assumptions on how position bias affects users, the neural click model, and IPS approach that we compare in this work.

A model of position bias. We begin by assuming a model of how position bias affects the click behavior of users. For this work, we resort to the prevalent model in unbiased learning-to-rank, the position-based model (PBM) [16, 39]. Let $P(Y = 1 \mid d, q)$ be the probability of a document d being relevant to a given search query q and $P(O = 1 \mid k)$ the probability of observing a document at rank $k \in K, K = \{1, 2, \ldots\}$; then we assume that clicks occur only on items that were observed and relevant:

$$\begin{aligned} P(C = 1 \mid d, q, k) &= P(O = 1 \mid k) \cdot P(Y = 1 \mid d, q) \\ c_{d,k} &= o_k \cdot y_d. \end{aligned} \tag{1}$$

For brevity, we use the short notation above for the rest of the paper and drop the subscript q in all of our formulas assuming that the document relevance y_d is always conditioned on the current query context.

A neural position-based click model. A neural click model directly mirrors the PBM user model introduced in the previous section in its architecture [7, 13, 22, 54]. We use a neural network g to estimate document relevance $\hat{y}_d$ from features x_d and estimate position bias $\hat{o}_k$ using a single parameter per rank denoted by $f(k)$. We use sigmoid activations and multiply the resulting probabilities:

$$\begin{aligned} \hat{c}_{d,k} &= \sigma(f(k)) \cdot \sigma(g(x_d)) \\ \hat{c}_{d,k} &= \hat{o}_k \cdot \hat{y}_d. \end{aligned} \tag{2}$$

A common choice to fit neural click models is the binary cross-entropy loss between predicted and observed clicks in the dataset [22, 23, 54–56]:

$$\mathcal{L}_{\text{pbm}}(\hat{y}, \hat{o}) = -\sum_{(d,k) \in D} c_{d,k} \cdot \log(\hat{y}_d \cdot \hat{o}_k) + (1 - c_{d,k}) \cdot \log(1 - \hat{y}_d \cdot \hat{o}_k). \tag{3}$$

A pointwise IPS model. Instead of predicting clicks, IPS directly predicts the document relevance $\hat{y}_d$ and assumes an estimation of the position bias $\hat{o}_k$ is given [28, 40]. Thus, the IPS model we assume in this work only uses the relevance network g:

$$\hat{y}_d = g(x_d). \tag{4}$$

Bekker et al. [5] introduce a pointwise IPS loss that minimizes the binary cross-entropy between predicted and true document relevance. Note how the PBM assumption is used to recover the unbiased document relevance by dividing clicks by the estimated position bias $\hat{o}_k$:

$$\mathcal{L}_{\text{ips}}(\hat{y}, \hat{o}) = -\sum_{(d,k) \in D} \frac{c_{d,k}}{\hat{o}_k} \cdot \log(\hat{y}_d) + \left(1 - \frac{c_{d,k}}{\hat{o}_k}\right) \cdot \log(1 - \hat{y}_d). \tag{5}$$

4 Methods

4.1 Comparing unbiasedness

In this section, we compare the ability of the neural click model and pointwise IPS ranker to recover the unbiased relevance of an item under position bias. We begin by noting that in the trivial case in which there is no position bias, i.e., clicks are an unbiased indicator of relevance, both approaches are identical.

Proposition 1. *When correctly assuming that no position bias exists, i.e.,* $\forall k \in K, o_k = \hat{o}_k = 1$*, the click model and pointwise IPS method are equivalent:*

$$\mathbb{E}\left[\mathcal{L}_{ips}(\hat{y}, \hat{o})\right] = \mathbb{E}\left[\mathcal{L}_{pbm}(\hat{y}, \hat{o})\right] = -\sum_{(d,k)\in D} y_d \cdot \log(\hat{y}_d) + (1 - y_d) \cdot \log(1 - \hat{y}_d).$$

Second, both approaches also collapse to the same (biased) model in the case of not correcting for an existing position bias in the data.

Proposition 2. *When falsely assuming that no position bias exists, i.e.,* $\forall k \in K, \hat{o}_k = 1 \wedge o_k < 1$*, the click model and pointwise IPS method are equivalently biased:*

$$\mathbb{E}\left[\mathcal{L}_{ips}(\hat{y}, \hat{o})\right] = \mathbb{E}\left[\mathcal{L}_{pbm}(\hat{y}, \hat{o})\right] = -\sum_{(d,k)\in D} y_d o_k \cdot \log(\hat{y}_d) + (1 - y_d o_k) \cdot \log(1 - \hat{y}_d).$$

However, how do both approaches compare when inferring the unbiased document relevance under an existing position bias? Saito et al. [40] show that $\mathcal{L}_{\text{ips}}(\hat{y})$ is unbiased if the position bias is correctly estimated, $\forall k \in K, \hat{o}_k = o_k$ and users actually behave according to the PBM [40, Proposition 4.3]). The notion of an unbiased estimator is harder to apply to neural click models, since relevance is a parameter to be inferred. Instead of unbiasedness, Oosterhuis [32] looks into consistency of click models and shows that click models jointly estimating both bias and relevance parameters are not consistent estimators of document relevance. This means that there are rankings in which even infinite click data will not lead to the true document relevance estimate.

But what happens if click models do not have to jointly estimate bias and relevance parameters, but only item relevance? Since IPS approaches often assume access to a correctly estimated position bias [1, 28, 34, 40, 45], we investigate this idealized setting for the click model and show that initializing the model parameters $\hat{o}_k$ with the true position bias leads to an unbiased relevance estimate.

Theorem 1. *The click model is an unbiased estimator of relevance when given access to the true position bias:*

$$\mathbb{E}\left[\hat{y}_d\right] = \frac{o_k y_d}{\hat{o}_k}, \forall k \in K, \hat{o}_k = o_k. \tag{6}$$

Proof. We begin by taking the partial derivative of $\mathcal{L}_{\text{pbm}}$ with regard to the estimated document relevance $\hat{y}$ in our click model. Since the model factorizes, for ease of notation we will look at a single document and single observation:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{pbm}}}{\partial \hat{y}} &= -\left(c \cdot \frac{\partial}{\partial \hat{y}}\left[\log(\hat{o}\hat{y})\right] + (1-c) \cdot \frac{\partial}{\partial \hat{y}}\left[\log(1-\hat{o}\hat{y})\right]\right) \\ &= -\left(c \cdot \frac{\hat{o}}{\hat{o}\hat{y}} + (1-c) \cdot \frac{-\hat{o}}{1-\hat{o}\hat{y}}\right) \\ &= -\left(\frac{c}{\hat{y}} + \frac{-\hat{o} + \hat{o}c}{1-\hat{o}\hat{y}}\right) \\ &= -\frac{c-\hat{o}\hat{y}}{\hat{y}(1-\hat{o}\hat{y})}.\end{aligned} \tag{7}$$

Next, we find the ideal model minimizing the loss by finding the roots of the derivative. We note that this function is convex and any extrema found will be a minimum:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{pbm}}}{\partial \hat{y}} &= 0 \\ -\frac{c-\hat{o}\hat{y}}{\hat{y}(1-\hat{o}\hat{y})} &= 0 \\ \hat{y} &= \frac{c}{\hat{o}}.\end{aligned} \tag{8}$$

Lastly, in expectation we see that the obtained relevance estimate is the true document relevance when the estimated and true position bias are equal:

$$\begin{aligned}\mathbb{E}\left[\hat{y}\right] &= \frac{\mathbb{E}\left[c\right]}{\hat{o}} \\ \mathbb{E}\left[\hat{y}\right] &= \frac{oy}{\hat{o}}.\end{aligned} \tag{9}$$

Thus, given the correct position bias, we find that both the click model and IPS objective optimize for the unbiased document relevance, suggesting a similar performance in an idealized benchmark setup. But before covering our empirical comparison, we want to note one additional difference of both loss functions.

4.2 A difference in loss magnitude

We note one difference between the click model and IPS-based loss functions concerning their magnitude and relationship with position bias. While IPS-based loss functions are known to suffer from high variance due to dividing clicks by potentially small probabilities [44, 51], the neural click model seems to suffer from the opposite problem since both $y_{d,k}$ and $\hat{y}_{d,k}$ (assuming our user model is correct) are multiplied by a potentially small examination probability. Thus, independent of document relevance, items at lower positions have a click probability closer to zero, impacting the magnitude of the loss (and gradient). Fig. 1 visualizes the loss for a single item of relevance $y_d = 0.5$ under varying degrees of position bias. While the pointwise IPS loss in expectation of infinite clicks always converges to the same distribution, the click model's loss gets smaller in magnitude with an increase in position bias. While the magnitude differs, the minimum of the loss, as shown earlier in Section 4.1, is still correctly positioned at 0.5. We will explore if this difference in loss magnitude might negatively impact items at lower positions in our upcoming experiments.

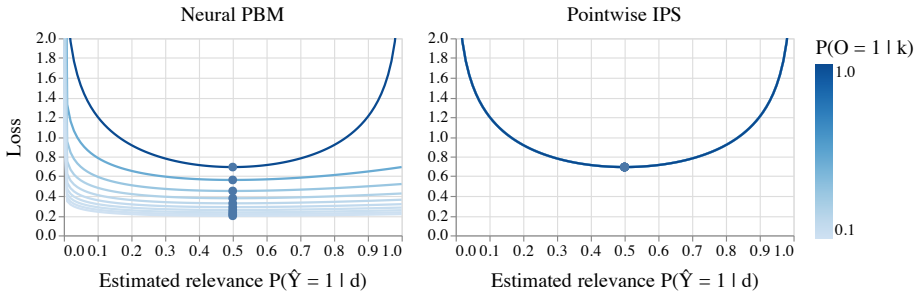


Fig. 1: Visualizing $\mathcal{L}_{\text{pbm}}$ on the left and $\mathcal{L}_{\text{ips}}$ on the right for a single document of relevance $y_d = 0.5$ under varying degrees of position bias.

Table 1: Overview of the LTR datasets used in this work.

Dataset	#Features	#Queries	%Train / val / test	#Documents per query				
				min	mean	med.	p90	max
MSLR-WEB30K	136	31,531	60 / 20 / 20	1	120	109	201	1,251
Istella-S	220	33,018	58.3 / 19.9 / 21.8	3	103	120	147	182
Yahoo! Webscope	699	29,921	66.6 / 10 / 23.3	1	24	19	49	139

5 Experimental setup

To compare click model and IPS-based approaches empirically, we use an evaluation setup that is prevalent in unbiased learning-to-rank [24, 26, 28, 33, 35, 36, 45, 47]. The main idea is to use real-world LTR datasets containing full expert annotations of item relevance to generate synthetic clicks according to our user model. Below, we describe the used datasets, the click generation procedure, as well as model implementation and training.

Datasets. We use three large-scale public LTR datasets to simulate synthetic user clicks: *MSLR-WEB30k* [37], *Istella-S* [17], and *Yahoo! Webscope* [9]. Each query-document pair is represented by a feature vector x_d and is accompanied by a score $s_d \in \{0, 1, 2, 3, 4\}$ indicating relevance as judged by a human annotator. Table 1 contains an overview of the dataset statistics. During preprocessing, we normalize the document feature vectors of *MSLR-WEB30k* and *Istella-S* using $\log1\text{p}(x_d) = \log_e(1 + |x_d|) \odot \text{sign}(x_d)$, as recently suggested by Qin et al. [38]. The features of *Yahoo! Webscope* come already normalized [9]. We use stratified sampling to limit each query to contain at most the 90th percentile number of documents (Table 1), improving computational speed while keeping the distribution of document relevance in the datasets almost identical.

Simulating user behavior. Our click simulation setup closely follows [45, 47]. First, we train a LightGBM [29] implementation of LambdaMART [8] on 20 sampled train queries with fully supervised relevance annotations as our production ranker.[3] The intuition is to simulate initial rankings that are better than random but leave room for further improvement.

[3] LightGBM Version 3.3.2, using 100 trees, 31 leafs, and learning rate 0.1.

We generate up to 100 million clicks on our train and validation sets by repeatedly: (i) sampling a query uniformly at random from our dataset; (ii) ranking the associated documents using our production ranker; and (iii) generating clicks according to the PBM user model (Eq. 1). As in [45], we generate validation clicks proportional to the train / validation split ratio in each dataset (Table 1). When sampling clicks according to the PBM, we use the human relevance labels provided by the datasets as ground truth for the document relevance y_d. We use a graded notion of document relevance [3, 4, 10, 25] and add click noise of $\epsilon = 0.1$ to also sample clicks on documents of zero relevance:

$$y_d = \epsilon + (1 - \epsilon) \cdot \frac{2^{s_d} - 1}{2^4 - 1}. \tag{10}$$

We follow Joachims et al. [28] and simulate the position bias for a document at rank k after preranking as:

$$o_k = \left(\frac{1}{k}\right)^{\eta} \tag{11}$$

The parameter η controls the strength of position bias; $\eta = 0$ corresponds to no position bias. We use a default of $\eta = 1$. Lastly, we apply an optimization step from [34] and train on the average click-through-rate of each query-document pair instead of the actual sampled raw click data [34, Eq. 39]. This allows us to scale our simulation to millions of queries and multiple repetitions while keeping the computational load almost constant. Our experimental results hold up without this trick.

Model implementation and training. We estimate document relevance from features using the same network architecture $g(x_d)$ for both the click model and IPS-based ranker. Similar to [45, 46], we use a three layer feed-forward network with $[512, 256, 128]$ neurons, ELU activations, and dropout 0.1 in the last two layers. We pick the best-performing optimizer[4] and learning rate[5] over five independent runs on the validation set for each model. In all experiments, we train our models on the synthetic click datasets up to 200 epochs and stop early after five epochs of no improvement of the validation loss. We do not clip propensities in the IPS model to avoid introducing bias [1, 28].

Experimental runs. We follow related work and report the final evaluation metrics on the original annotation scores of the test set [1, 28, 34]. We test differences for significance using a two-tailed student's t-test [43], apply the Bonferroni correction [6] to account for multiple comparisons, and use a significance level of $\alpha = 0.0001$. All results reported in this work are evaluated over ten independent simulation runs with different random seeds. We compare five models:

IPS / PBM - Naive: A version of both models that does not compensate for position bias. In this case both models are equivalent (Proposition 2).

IPS - True bias: Pointwise IPS ranker with access to the true simulated position bias.

PBM - Estimated bias: Neural PBM jointly inferring position bias and document relevance during training.

[4] optimizer $\in \{Adam, Adagrad, SGD\}$

[5] learning rate $\in \{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$

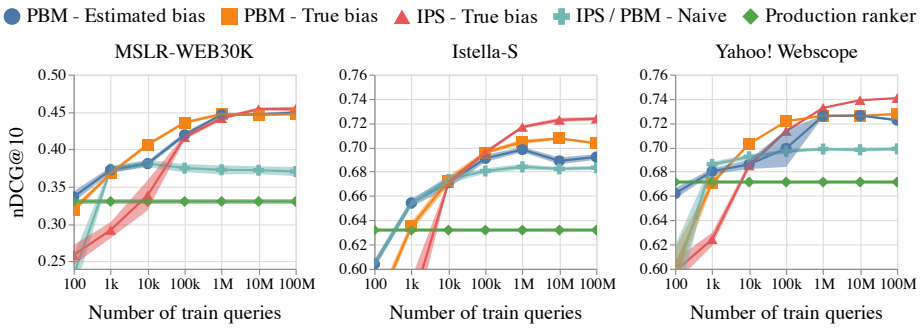


Fig. 2: Test performance after training on up to 100M simulated queries. All results are averaged over ten independent runs, and we display a bootstrapped 95% confidence interval.

PBM - True bias: Neural PBM initialized with the true position bias; the bias is fixed during training.

Production ranker: LambdaMART production ranker used to pre-rank queries during simulation.

6 Results and analysis

We examine if the neural click model and pointwise IPS models are empirically equivalent in a semi-synthetic click simulation.

6.1 Main findings

Fig. 2 displays the test performance of all model combinations when training up to 100M simulated queries; full tabular results are available in Table 2. Inspecting Fig. 2, we first note that all approaches improve over the initial rankings provided by the production ranker. The version of both models not correcting for position bias (*IPS / PBM - Naive*) converges to its final, suboptimal, performance after one million clicks. Significantly improving over the naive baseline on two out of three datasets (except *Istella-S*) is the neural click model jointly estimating position bias and relevance (*PBM - Estimated bias*).

Next, we see that providing the *PBM - True Bias* model with access to the correct position bias stabilizes and improves performance significantly over the naive baseline on all datasets. While having a lower variance, the improvements over *PBM - Estimated Bias* are not significant on any of the datasets. The *IPS - True bias* model is less effective than the neural click models for the first 100k clicks but ends up outperforming the click model significantly on two of the three LTR datasets (*Istella-S* and *Yahoo! Webscope*). These differences under idealized conditions between pointwise IPS and the click model are small, but significant. And to our surprise, the neural click model performs worse than the pointwise IPS model, even with access to the true position bias.

In Theorem 1, we prove that click models can recover unbiased document relevance when the position bias is accurately estimated. However, our empirical

Table 2: Ranking performance on the full-information test set after 100M train queries as measured in nDCG and Average Relevant Position (ARP) [28]. Results are averaged over ten independent runs, displaying the standard deviation in parentheses. We mark significantly higher ▲ or lower performance ▼ compared to the **PBM - True bias** model using a significance level of $\alpha = 0.0001$.

Dataset	Model	nDCG@5 ↑	nDCG@10 ↑	ARP ↓
MSLR-WEB30K	Production	0.301 (0.027) ▼	0.330 (0.024) ▼	49.223 (0.693) ▲
	Naive	0.348 (0.022) ▼	0.370 (0.020) ▼	48.386 (0.538) ▲
	PBM - Est. Bias	0.429 (0.010)	0.449 (0.008)	44.835 (0.274)
	PBM - True Bias	0.428 (0.006)	0.447 (0.006)	44.965 (0.230)
	IPS - True Bias	0.432 (0.011)	0.454 (0.010)	44.418 (0.227)
Istella-S	Production	0.566 (0.012) ▼	0.632 (0.010) ▼	10.659 (0.207) ▲
	Naive	0.616 (0.005) ▼	0.683 (0.005) ▼	9.191 (0.154) ▲
	PBM - Est. Bias	0.629 (0.008)	0.692 (0.007)	10.605 (1.193)
	PBM - True Bias	0.638 (0.003)	0.703 (0.004)	8.911 (0.212)
	IPS - True Bias	0.656 (0.005) ▲	0.724 (0.004) ▲	8.274 (0.141) ▼
Yahoo! Webscope	Production	0.613 (0.012) ▼	0.671 (0.009) ▼	10.439 (0.095) ▲
	Naive	0.647 (0.006) ▼	0.699 (0.004) ▼	10.199 (0.052) ▲
	PBM - Est. Bias	0.673 (0.005)	0.722 (0.003)	9.848 (0.055)
	PBM - True Bias	0.680 (0.004)	0.728 (0.003)	9.812 (0.035)
	IPS - True Bias	0.695 (0.001) ▲	0.741 (0.001) ▲	9.658 (0.011) ▼

evaluation indicates a difference between click model and IPS-based approaches, even under the idealized conditions assumed in this setup: *unlike the IPS-based approach, the neural click model may suffer from bias.* Given this observed difference, we conduct further analyses by revisiting the effect of position bias on the magnitude of the click model's loss discussed earlier in Section 4.2.

6.2 Further analyses

Our first hypothesis to explain the lower performance of the neural click model concerns hyperparameter tuning. Section 4.2 shows that the click model loss decreases with an increase in position bias. Through manual verification, we find that items at lower positions have smaller gradient updates, affecting the choice of learning rate and the number of training epochs. While this is a concern when using SGD, our extensive hyperparameter tuning and use of adaptive learning rate optimizers should mitigate this issue (Section 5). Hence, we reject this hypothesis.

Instead, we hypothesize that higher ranked items might overtake the gradient of lower ranked items, given their higher potential for loss reduction. This case might occur when encountering two documents with similar features but different relevance. The item at the higher position could bias the expected relevance towards its direction. This is indeed what we find when simulating a toy scenario with two documents in Figure 3. There, we display one relevant but rarely observed document (red triangle) and one irrelevant but always observed item (orange square). Both click model and IPS approaches converge to the correct

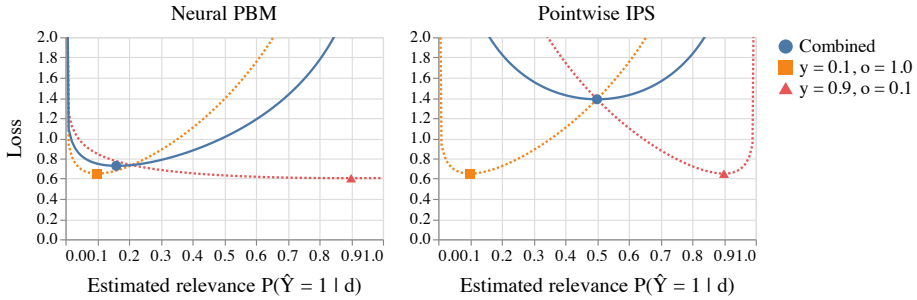
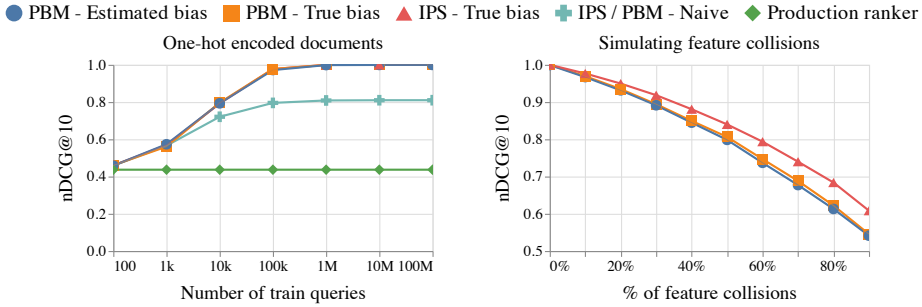


Fig. 3: Visualizing the loss and estimated document relevance of two documents when calculated separately (dotted lines) and combined (solid line).



(a) Test performance when documents share no features.

(b) Gradually introducing feature collisions between documents.

Fig. 4: Experiments on one-hot encoded documents. All results are averaged over ten independent runs. We display a bootstrapped 95% confidence interval.

document relevance when computing the loss for each item separately, but when computing the combined average loss, the IPS approach converges to the mean relevance of both items while the click model is biased towards the item with the higher examination probability.

One can frame this finding as an instance of *model misfit*. Theorem 1 demands a separate parameter $\hat{y}_d$ for each query-document pair, but by generalizing over features using the relevance network g, we might project multiple documents onto the same parameter $\hat{y}_d$, which might be problematic when features do not perfectly capture item relevance. We test our hypothesis that the click model's gradient updates are biased towards items with higher examination probabilities with three additional experiments.

No shared document features. First, we should see an equivalent performance of both approaches in a setting in which documents share no features since the gradient magnitude should not matter in this setting. We create a fully synthetic dataset of 10,000 one-hot encoded vectors with uniform relevance scores between 0 and 4. To avoid feature interactions, we reduce the depth of the relevance network g to a single linear layer. We find in Fig. 4a that indeed both approaches are able to recover the true document relevance. Every document in the validation or test set appears once in the train dataset, thus achieving a perfect ranking score (e.g., $nDCG@10 = 1.0$) is possible in this setting.

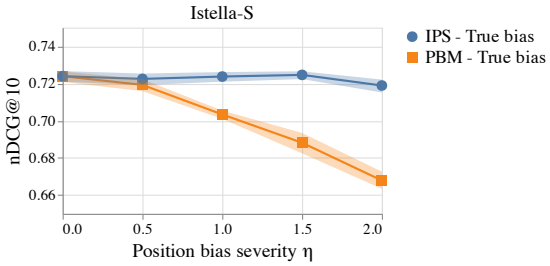


Fig. 5: Simulating an increasing (known) position bias. We report test performance after 100M clicks over 10 independent runs.

Feature collisions. Second, gradually forcing documents to share features by introducing random feature collisions into our one-hot encoded dataset should lead to a stronger drop in performance for the click model. At the start of each simulation, we use a modulo operation to assign a share of documents based on their id on to the same one-hot encoded feature vectors. Fig. 4b shows that both approaches perform equivalently when each document has its own feature vector. But when gradually introducing collisions, *PBM - Estimated bias* and *PBM - True bias* deteriorate faster in performance than *IPS - True bias*.

Mitigating position bias. A last interesting consequence is that this problem should get worse with an increase in (known) position bias. Simulating an increasing position bias and supplying the examination probabilities to both approaches on *Istella-S* shows that IPS can recover consistently from high position bias, while the click model deteriorates in performance with an increase in position bias (Fig. 5).

In summary, we found strong evidence that when encountering documents of different relevance but similar features, the neural click model biases its relevance estimate towards items with higher exposure.

7 Conclusion

We have considered whether recent neural click models and pointwise IPS rankers are equivalent for pointwise learning-to-rank from position-biased user clicks. We show theoretically and empirically that neural click models and pointwise IPS rankers achieve equal performance when the true position bias is known, and relevance is estimated for each item separately. However, we also find small but significant empirical differences, indicating that the neural click model may be affected by position bias when learning from shared and potentially conflicting document features.

Given the similarity of the neural PBM used in this work to current industry trends [22, 23, 54, 55], we urge practitioners to investigate if their model architecture is vulnerable to the described bias, especially when representing items using a small set of features or low dimensional latent embeddings. Potential diagnostic tools include simulating synthetic clicks or training a related pointwise IPS method to test for performance improvements.

We emphasize that our findings are specific to our neural PBM setup, and we make no claims about other architectures, such as additive two-tower models [54] or click models trained using expectation maximization [50]. We plan to further investigate connections and differences between IPS and click models, extending our evaluation beyond the pointwise setting to more sophisticated conditions such as mixtures of user behavior and bias misspecification. We share our code at https://github.com/philipphager/ultr-cm-vs-ips/

Acknowledgements We thank our reviewers for their time and valuable feedback. For insightful discussions and their comments, we thank Shashank Gupta, Romain Deffayet, Kathrin Parchatka, and Harrie Oosterhuis.

This research was supported by the Mercury Machine Learning Lab, a collaboration between TU Delft, the University of Amsterdam, and Booking.com. Maarten de Rijke was supported by the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl.

All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

Bibliography

[1] Agarwal, A., Takatsu, K., Zaitsev, I., Joachims, T.: A general framework for counterfactual learning-to-rank. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (2019)

[2] Agarwal, A., Zaitsev, I., Wang, X., Li, C., Najork, M., Joachims, T.: Estimating position bias without intrusive interventions. In: International Conference on Web Search and Data Mining (WSDM) (2019)

[3] Ai, Q., Bi, K., Luo, C., Guo, J., Croft, W.B.: Unbiased learning to rank with unbiased propensity estimation. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (2018)

[4] Ai, Q., Yang, T., Wang, H., Mao, J.: Unbiased learning to rank: Online or offline? ACM Transactions on Information Systems (TOIS) **39**(2) (2021)

[5] Bekker, J., Robberechts, P., Davis, J.: Beyond the selected completely at random assumption for learning from positive and unlabeled data. In: Machine Learning and Knowledge Discovery in Databases: European Conference (ECML PKDD) (2019)

[6] Bonferroni, C.: Teoria statistica delle classi e calcolo delle probabilita. Pubblicazioni del R. Istituto Superiore di Scienze Economiche e Commericiali di Firenze **8**, 3–62 (1936)

[7] Borisov, A., Markov, I., de Rijke, M., Serdyukov, P.: A neural click model for web search. In: The World Wide Web Conference (WWW) (2016)

[8] Burges, C.J.: From ranknet to lambdarank to lambdamart: An overview. Tech. Rep. MSR-TR-2010-82, Microsoft (2010)

[9] Chapelle, O., Chang, Y.: Yahoo! learning to rank challenge overview. Journal of Machine Learning Research (JMLR) **14**, 1–24 (2011)

[10] Chapelle, O., Metlzer, D., Zhang, Y., Grinspan, P.: Expected reciprocal rank for graded relevance. In: International Conference on Information and Knowledge Management (CIKM) (2009)

[11] Chapelle, O., Zhang, Y.: A dynamic bayesian network click model for web search ranking. In: The World Wide Web Conference (WWW) (2009)

[12] Chen, J., Mao, J., Liu, Y., Zhang, M., Ma, S.: A context-aware click model for web search. In: International Conference on Web Search and Data Mining (WSDM) (2020)

[13] Chu, W., Li, S., Chen, C., Xu, L., Cui, H., Liu, K.: A general framework for debiasing in ctr prediction (2021), https://doi.org/10.48550/arXiv.2112.02767

[14] Chuklin, A., Markov, I., de Rijke, M.: Click Models for Web Search. Morgan & Claypool (2015), ISBN 9781627056489, https://doi.org/10.2200/S00654ED1V01Y201507ICR043

[15] Covington, P., Adams, J., Sargin, E.: Deep neural networks for youtube recommendations. In: ACM Conference on Recommender Systems (RecSys) (2016)

[16] Craswell, N., Zoeter, O., Taylor, M., Ramsey, B.: An experimental comparison of click position-bias models. In: International Conference on Web Search and Data Mining (WSDM) (2008)

[17] Dato, D., Lucchese, C., Nardini, F.M., Orlando, S., Perego, R., Tonellotto, N., Venturini, R.: Fast ranking with additive ensembles of oblivious and non-oblivious regression trees. ACM Transactions on Information Systems (TOIS) **35**(2) (2016)
[18] Diaz, F., White, R., Buscher, G., Liebling, D.: Robust models of mouse movement on dynamic web search results pages. In: International Conference on Information and Knowledge Management (CIKM) (2013)
[19] Dupret, G.E., Piwowarski, B.: A user browsing model to predict search engine click data from past observations. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (2008)
[20] Gomez-Uribe, C.A., Hunt, N.: The netflix recommender system: Algorithms, business value, and innovation. ACM Transactions on Management Information Systems (TMIS) **6**(4) (2016)
[21] Guo, F., Liu, C., Kannan, A., Minka, T., Taylor, M., Wang, Y.M., Faloutsos, C.: Click chain model in web search. In: The World Wide Web Conference (WWW) (2009)
[22] Guo, H., Yu, J., Liu, Q., Tang, R., Zhang, Y.: Pal: A position-bias aware learning framework for ctr prediction in live recommender systems. In: ACM Conference on Recommender Systems (RecSys) (2019)
[23] Haldar, M., Ramanathan, P., Sax, T., Abdool, M., Zhang, L., Mansawala, A., Yang, S., Turnbull, B., Liao, J.: Improving deep learning for airbnb search. In: International Conference on Knowledge Discovery and Data Mining (SIGKDD) (2020)
[24] Hofmann, K., Schuth, A., Whiteson, S., de Rijke, M.: Reusing historical interaction data for faster online learning to rank for ir. In: International Conference on Web Search and Data Mining (WSDM) (2013)
[25] Hu, Z., Wang, Y., Peng, Q., Li, H.: Unbiased lambdamart: An unbiased pairwise learning-to-rank algorithm. In: The World Wide Web Conference (WWW) (2019)
[26] Jagerman, R., Oosterhuis, H., de Rijke, M.: To model or to intervene: A comparison of counterfactual and online learning to rank from user interactions. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (2019)
[27] Joachims, T., Granka, L., Pan, B., Hembrooke, H., Gay, G.: Accurately interpreting clickthrough data as implicit feedback. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (2005)
[28] Joachims, T., Swaminathan, A., Schnabel, T.: Unbiased learning-to-rank with biased feedback. In: International Conference on Web Search and Data Mining (WSDM) (2017)
[29] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. In: International Conference on Neural Information Processing Systems (NIPS) (2017)
[30] Lin, J., Liu, W., Dai, X., Zhang, W., Li, S., Tang, R., He, X., Hao, J., Yu, Y.: A graph-enhanced click model for web search. In: International ACM

SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (2021)
[31] Liu, T.Y., et al.: Learning to rank for information retrieval. Foundations and Trends in Information Retrieval **3**, 225–331 (2009)
[32] Oosterhuis, H.: Reaching the end of unbiasedness: Uncovering implicit limitations of click-based learning to rank. In: International Conference on the Theory of Information Retrieval (ICTIR) (2022)
[33] Oosterhuis, H., de Rijke, M.: Differentiable unbiased online learning to rank. In: International Conference on Information and Knowledge Management (CIKM) (2018)
[34] Oosterhuis, H., de Rijke, M.: Policy-aware unbiased learning to rank for top-k rankings. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (2020)
[35] Oosterhuis, H., de Rijke, M.: Unifying online and counterfactual learning to rank: A novel counterfactual estimator that effectively utilizes online interventions. In: International Conference on Web Search and Data Mining (WSDM) (2021)
[36] Ovaisi, Z., Ahsan, R., Zhang, Y., Vasilaky, K., Zheleva, E.: Correcting for selection bias in learning-to-rank systems. In: The Web Conference (2020)
[37] Qin, T., Liu, T.: Introducing letor 4.0 datasets (2013), `https://doi.org/10.48550/arXiv.1306.2597`
[38] Qin, Z., Yan, L., Zhuang, H., Tay, Y., Pasumarthi, R.K., Wang, X., Bendersky, M., Najork, M.: Are neural rankers still outperformed by gradient boosted decision trees? In: International Conference on Learning Representations (ICLR) (2021)
[39] Richardson, M., Dominowska, E., Ragno, R.: Predicting clicks: Estimating the click-through rate for new ads. In: The World Wide Web Conference (WWW) (2007)
[40] Saito, Y., Yaginuma, S., Nishino, Y., Sakata, H., Nakata, K.: Unbiased recommender learning from missing-not-at-random implicit feedback. In: International Conference on Web Search and Data Mining (WSDM) (2020)
[41] Sanderson, M., et al.: Test collection based evaluation of information retrieval. Foundations and Trends in Information Retrieval **4**, 247–375 (2010)
[42] Sorokina, D., Cantu-Paz, E.: Amazon search: The joy of ranking products. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (2016)
[43] Student: The probable error of a mean. Biometrika pp. 1–25 (1908)
[44] Swaminathan, A., Joachims, T.: The self-normalized estimator for counterfactual learning. In: International Conference on Neural Information Processing Systems (NIPS) (2015)
[45] Vardasbi, A., Oosterhuis, H., de Rijke, M.: When inverse propensity scoring does not work: Affine corrections for unbiased learning to rank. In: International Conference on Information and Knowledge Management (CIKM) (2020)
[46] Vardasbi, A., de Rijke, M., Markov, I.: Cascade model-based propensity estimation for counterfactual learning to rank. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (2020)

[47] Vardasbi, A., de Rijke, M., Markov, I.: Mixture-Based Correction for Position and Trust Bias in Counterfactual Learning to Rank (2021)
[48] Wang, C., Liu, Y., Wang, M., Zhou, K., Nie, J.y., Ma, S.: Incorporating non-sequential behavior into click models. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (2015)
[49] Wang, X., Bendersky, M., Metzler, D., Najork, M.: Learning to rank with selection bias in personal search. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (2016)
[50] Wang, X., Golbandi, N., Bendersky, M., Metzler, D., Najork, M.: Position bias estimation for unbiased learning to rank in personal search. In: International Conference on Web Search and Data Mining (WSDM) (2018)
[51] Wang, Y.X., Agarwal, A., Dudík, M.: Optimal and adaptive off-policy evaluation in contextual bandits. In: International Conference on Machine Learning (ICML) (2017)
[52] Xie, X., Liu, Y., Wang, X., Wang, M., Wu, Z., Wu, Y., Zhang, M., Ma, S.: Investigating examination behavior of image search users. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (2017)
[53] Xie, X., Mao, J., de Rijke, M., Zhang, R., Zhang, M., Ma, S.: Constructing an interaction behavior model for web image search. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (2018)
[54] Yan, L., Qin, Z., Zhuang, H., Wang, X., Bendersky, M., Najork, M.: Revisiting two-tower models for unbiased learning to rank. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (2022)
[55] Zhao, Z., Hong, L., Wei, L., Chen, J., Nath, A., Andrews, S., Kumthekar, A., Sathiamoorthy, M., Yi, X., Chi, E.: Recommending what video to watch next: A multitask ranking system. In: ACM Conference on Recommender Systems (RecSys) (2019)
[56] Zhuang, H., Qin, Z., Wang, X., Bendersky, M., Qian, X., Hu, P., Chen, D.C.: Cross-positional attention for debiasing clicks. In: The Web Conference (2021)