Propensity-Independent Bias Recovery in Offline Learning-to-Rank Systems

Zohreh Ovaisi
zovais2@uic.edu
University of Illinois at Chicago

Kathryn Vasilaky
kvasilak@calpoly.edu
California Polytechnic State University

Elena Zheleva
ezheleva@uic.edu
University of Illinois at Chicago

ABSTRACT

Learning-to-rank systems often utilize user-item interaction data (e.g., clicks) to provide users with high-quality rankings. However, this data suffers from several biases, and if naively used as training data, it can lead to suboptimal ranking algorithms. Most existing bias-correcting methods focus on position bias, the fact that higher-ranked results are more likely to receive interaction, and address this bias by leveraging inverse propensity weighting. However, it is not always possible to accurately estimate propensity scores, and in addition to position bias, selection bias is often encountered in real-world recommender systems. Selection bias occurs because users are exposed to a truncated list of results, which gives a zero chance for some items to be observed and, therefore, interacted with, even if they are relevant. Here, we propose a new counterfactual method that uses a two-stage correction approach and jointly addresses selection and position bias in learning-to-rank systems without relying on propensity scores. Our experimental results show that our method is better than state-of-the-art propensity-independent methods and either better than or comparable to methods that make the strong assumption for which the propensity model is known.

CCS CONCEPTS

• **Information systems** → **Learning to rank**.

KEYWORDS

recommender systems; position bias; selection bias

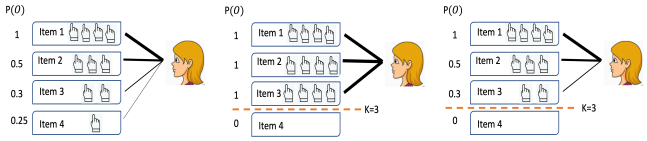
ACM Reference Format:
Zohreh Ovaisi, Kathryn Vasilaky, and Elena Zheleva. 2021. Propensity-Independent Bias Recovery in Offline Learning-to-Rank Systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3463097

1 INTRODUCTION

Recommender and learning-to-rank (LTR) systems play a central role in how online technologies impact our daily lives, from deciding what products to buy to which news to read and what movies to see.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3463097



(a) Position bias **(b) Selection bias** **(c) Position bias and selection bias**

Figure 1: Position and selection bias effect. P(O) represents observation probability which decreases from top to bottom.

The main goal of a recommender system is to infer the relevance of items given user queries and then show users more relevant items. To this end, LTR systems are generally trained on either explicit relevance data (e.g., rating or human-annotated data) or implicit relevance data (e.g., clicks). While explicit relevance data is considered to be a more reliable signal of true relevance, it is often expensive to collect on a large scale. Thus, LTR systems generally use implicit feedback to estimate the relevance of each item given a query. However, implicit feedback is subject to several biases that, unless accounted for, lead to invalid inferences and poor ranking.

Much of the previous work on unbiased LTR systems has focused on position bias [1, 4, 7, 12–14, 17, 18, 20, 23, 29, 30, 33, 34, 38], the fact that users are more likely to interact with higher-ranked results (Figure 1a). A well-known solution for tackling position bias is using an Inverse Propensity Score (IPS) approach [4, 5, 8, 9, 13, 15–17, 24, 26, 33, 34]. Propensity score is the probability of an item being observed given its position, and IPS techniques re-weight the relevance score of the item using the inverse of its propensity score. Debiasing methods that require propensity knowledge share some common drawbacks. First, even though proper propensity estimation can be achieved through online result randomization [1, 17] where the same item is shown in multiple positions for the same query, this experimentation can negatively impact user experience [4, 34]. Second, estimating propensities by integrating click logs from multiple ranking functions [3, 8]. assumes that the same set of query-document pairs appear in multiple positions which may be unrealistic, especially for long-tail queries, resulting in misspecified propensity estimation [31]. Finally, propensities can be learned jointly with ranking models using offline click data [4, 13, 34, 39] but this implies that this approach can control neither for the relevance nor for the bias, leading to biased estimates [3, 31].

Propensity-dependent LTR algorithms generally assume that all items have a non-zero probability of being observed and, therefore, being clicked, and weight only clicked items in the empirical loss function. However, many items have a zero probability of being observed and clicked either because the ranking system displays

only the top-k items, or because users do not peruse all the listed results. This introduces selection bias [14, 21] (Figure 1b), and LTR algorithms that do not account for it it will exclude such items from future training data even if they are relevant [21, 31]. Related work that accounts only for selection bias [11, 27, 28, 35, 36] assumes that all shown items have an equal chance of interaction [28, 35]. However, the top-k items have different chance of being clicked depending on their positions, whereas any remaining tail items have no chance (Figure 1c). Ignoring position and selection bias propagates these biases further and degrades a platform's utility.

There are few methods that address both biases. Oosterhuis and de Rijke [19] propose a policy-aware propensity estimator which requires the stochastic logging policy to be known in advance. Yuan et al. [37] and Saito [25] introduce doubly robust LTR frameworks that combine a data imputation-based model with an IPS-based model, which only perform well if either the imputed error or the propensity score are accurately estimated. Ovaisi et al. [21] propose a counterfactual ensemble approach which incorporates a Heckman bivariate correction [10] for the selection bias component and *Propensity SVM*^{rank} [17] for the position bias component.

Here, we propose Propensity-Independent Joint Debiasing (PIJD), a two-stage bivariate method that is capable of jointly correcting for both position and selection bias without relying on propensity knowledge. We also develop a causal model that corresponds to how clicks are generated under these biases and show that the validity of our method is supported by that model. Our experiments demonstrate the value of PIJD in learning to rank from biased data.

2 PROBLEM DESCRIPTION

LTR systems learn to rank new items given a query x by modeling item relevance using past clicks. User engagement with an item y is treated as a signal of the relevance of that item given a query, where the relevance is modeled through the <query-item> features denoted as $\boldsymbol{F}_{x,y}$. The user-item interaction data is typically log data from a previously implemented recommender system. We refer to a recommender system that shows a ranking $\bar{\mathbf{y}}$ for a set of items, given a query as the base ranker S_{base}, and to the recommender system that learns to rank new items as the target ranker S.

2.1 Position and selection bias in ranked data

Data generated by recommender systems inherit the properties of their base ranker. Users are more likely to select items that are placed in high positions by a base ranker (position bias), while items truncated by the base ranker are discarded from ranking $\bar{\mathbf{y}}$, and, therefore, have a zero probability of selection in future LTR systems (selection bias). Given a ranking $\bar{\mathbf{y}}$ of items for query x produced by a ranker S_{base} and the rank of a particular item y in that ranking, $rank_{y,\bar{\mathbf{y}}}$, the probability of an item to be observed under position and selection bias can be modeled as:

$$P(o_{x,y} = 1|x, \bar{\mathbf{y}}) = \begin{cases} f(x, y, rank_{y,\bar{\mathbf{y}}}, \eta), & \text{if } rank_{y,\bar{\mathbf{y}}} < k \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where η is a parameter that captures the severity of position bias and f is a function that reflects the inverse relationship between the probability of being observed, $P(o_{x,y})$, and $rank_{y,\bar{\mathbf{y}}}$, as well as η.

2.2 Learning to rank

The goal of the LTR system is to find a target ranker $S \subset \mathcal{S}$ that minimizes the empirical risk $\hat{R}(S)$ on a sample of i.i.d. queries $\mathbf{x}$:

$$\hat{R}(S) = \frac{1}{|\mathbf{x}|} \sum_{x_i \in \mathbf{x}} \Delta\left(S(x_i)|x_i\right). \quad (2)$$

where $\Delta\left(S(x_i)|x_i\right)$ corresponds to the loss of ranking $\mathbf{y}_i$ produced by the target ranker S given a query x_i [17]. The loss function penalizes a ranking when relevant items appear low in the ranking. It does this by either comparing each ranking of ranker S to the true relevance of items when available (*full observation setting*) or to the noisy and biased relevance of items inferred from clicks in observational data (*partial observation setting*).

2.3 Learning to rank with biased data

The goal of this paper is to build an algorithm that learns to rank from biased data by jointly accounting for position and selection bias in the partial observation setting. Similar to previous work [17, 19, 21], we formulate the problem in a counterfactual framework [22].

Here, $O(x, y) \in \{0, 1\}$, is a variable indicating whether a user observed item y under query x, and $C_{O=1}(x, y) \in \{0, 1\}$ is the *click counterfactual* indicating whether an item y would have been clicked had y been observed. The goal of *ranking with counterfactuals* is to reliably estimate the probability of click counterfactuals for all items, including those that were never observed and clicked (i.e. y's with $O(x, y) = 0$ and $C(x, y) = 0$):

$$P(C_{O=1} = 1|rank_{y,\mathbf{y}} = i, \textbf{cutoff} = k, X = x, Y = y) \quad (3)$$

and then rank the items according to these probabilities. Here, X and Y are random variables corresponding to the query and the item. Solving the ranking within a counterfactual framework allows us to discover a target ranker S that returns ranking $S(x)$ for query x that is robust to selection and position bias.

While this sounds straightforward in theory, in practice we rarely know when a user observed an item unless it was clicked. A non-click can be a signal for "observing an item but not finding it relevant" or for "not observing the item at all" due to truncation of the item list. $O(x, y)$ depends both on whether the system chose to show item y to the user and the item's position.

3 PROPENSITY-INDEPENDENT JOINT-DEBIASING RANKING ALGORITHM

3.1 Causal model for LTR

We first describe a causal model for the click-generation process through which we justify our debiasing method and the identifiability of click counterfactuals. Following the notation of Ovaisi et al. [21], we define random variables used in our model: $S_{x,y}$ represents whether item y is shown ($S_{x,y} \in \{0, 1\}$), $O_{x,y}$ represents whether item y is observed by the user under query x ($O_{x,y} \in \{0, 1\}$), $C_{x,y}$ represents whether item y under query x is clicked ($C_{x,y} \in \{0, 1\}$, k represents the cutoff below which documents are not observed, $\boldsymbol{F}_{x,y}$ represents the features for each $< query, item >$ pair, $Rank_{y,\bar{\mathbf{y}}}$ represents the position of item y in the base ranking $\bar{\mathbf{y}}$, and $R_{x,y}$ represents y's true relevance score, which is latent.

Figure 2 is a causal graphical model that represents the position and selection bias in LTR systems. Given a query x, the base ranker

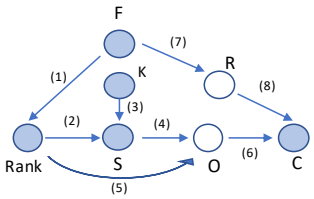


Figure 2: Causal model showing selection and position bias.

will rank item y based on the $< query, item >$ features $\boldsymbol{F}_{x,y}$ and place item y at $Rank_y$ (1). Because only the top k ranked items are shown to the user, $Rank_y$ and cutoff k impact whether an item is shown $S_{x,y}$ (2, 3). $S_{x,y}$ (4) and the item's ranking, $Rank_y$ (5), determine whether an item is observed ((4) and (5) are the sources of selection bias and position bias, respectively). Finally, whether an item is observed $O_{x,y}$ (6) as well as how relevant it is ($R_{x,y}$) (8) affect whether it is clicked. [1]

We are interested in the probability of a click had y been observed, which is identifiable from observational data when $O_{x,y}$ is known because $\mathbf{F}$ satisfies the backdoor criterion [22] relative to (O,C) in our causal model: $P(C_{O=1} = 1) = \sum_{\mathbf{f}\in\mathcal{F}} P(C_{O=1} = 1|\mathbf{F} = \mathbf{f})P(\mathbf{f}) = \sum_{\mathbf{f}\in\mathcal{F}} P(C = 1|\mathbf{F} = \mathbf{f}, O = 1)P(\mathbf{f})$. However, in practice, $O_{x,y}$ is latent and this formula implies looking at all possible feature combinations $\mathcal{F}$, which is unrealistic and would require a very large unbiased sample. Next, we discuss a method for estimating it.

3.2 PIJD: A Two-stage joint debiasing method

A simple way to model an item click is through the item relevance to the query based on the query and item features:

$$C_{x,y} = \boldsymbol{\alpha}\boldsymbol{F}_{x,y} + \epsilon_{x,y}. \tag{4}$$

where $\epsilon_{x,y}$ is a normally distributed error term. However, as shown in [21], this leads to biased estimation of the regression coefficients, $\boldsymbol{\alpha}$, because it does not take into consideration the fact that only certain items are represented in the data, and anything clicked is conditional on it having been observed by the user.

We address the selection and position biases in ranking data through a two-stage bivariate selection model in which item clicks and item observations are each modeled separately. Two-stage bivariate selection models are econometric models that deal with selection bias [10] and have been adapted to address selection bias in ranking data [21, 35] but have not been used for joint debiasing of position and selection bias. A two-stage Heckman model estimates the probability of selection into a sample, in our case the probability of an item being observed, and then controls for that probability in estimating features' effects on an outcome. The first stage of our Propensity-Independent Joint Debiasing (PIJD) model is:

$$O_{x,y} = \mathbb{1}_{\{\boldsymbol{\theta}\boldsymbol{F}_{x,y} + \theta_r f(Rank_{y,\bar{y}},k) + \epsilon^{(1)}_{x,y} \geq 0\}}. \tag{5}$$

This item observation process is estimated using a Probit model with $\boldsymbol{F}_{x,y}$ and $f(Rank_{y,\bar{y}}, k)$ as predictors where $f(Rank_{y,\bar{y}}, k)$ reflects both the direct and indirect effect of $Rank_{y,\bar{y}}$ on $O_{x,y}$. In practice, since $O_{x,y}$ is unknown, it is modeled as "possibly observed" and it equals 1 whenever $S_{x,y} = 1$. The estimated selection model is then used to create an Inverse Mills Ratio (IMR) for each $< query, item >$ pair, $\lambda_{x,y} = \frac{\phi(\hat{\boldsymbol{\theta}}\boldsymbol{F}_{x,y} + \hat{\theta}_r Rank_{y,\bar{y}})}{\Phi(\hat{\boldsymbol{\theta}}\boldsymbol{F}_{x,y} + \hat{\theta}_r Rank_{y,\bar{y}})}$, where $\phi()$ is the probability density function, $\Phi()$ is the cumulative distribution function for a standard normally distributed random variable, and $\hat{\boldsymbol{\theta}}$ is the $\boldsymbol{\theta}$ estimate obtained from Equation 5. Intuitively, the ratio represents the probability that a document is observed for a given $< query, item >$ pair relative to the cumulative probability of that document being observed across all queries. As such, it controls for the degree to which observing a document influences a user's decision to click on that document. Because it is computed based on *both* documents that are and are not selected by the algorithm - accounting for negative feedback - it is different from propensity scores that depend entirely on clicked documents. Clicks are user-driven and sparse [25], which subjects propensity estimation to misspecification.

The IMR is used in the second stage as a feature in Equation 4:

$$C_{x,y} = \boldsymbol{\alpha}^{unbiased}\boldsymbol{F}_{x,y} + \sigma\hat{\boldsymbol{\lambda}}_{x,y}(\hat{\boldsymbol{\theta}}\boldsymbol{F}_{x,y}, \theta_r Rank_{y,\bar{y}}) + \epsilon_{x,y} \tag{6}$$

where $\lambda_{x,y}$ is now conditioned on an item's position and selection bias. After completing the two stages, we can predict the click probabilities, $\hat{c}$, for each item given a query and then calculate item rankings based on these predicted probabilities. This two-stage setup now accounts for the truncation of data, which depends on the features of an item *and* its position. Note that IMR is calculated for both click and none click documents, and does not remove potentially relevant documents that were truncated.

Statistically, the IMR controls for the fact that the error terms in Equations 5 and 6 are not independent, and the click data, $C_{x,y}$, is not missing at random ([10][pg 155]). Assuming bivariate normality of the error terms, which OLS and Probit models satisfy, [10] shows that conditional on the IMR, estimates of α will be unbiased and normally distributed. It is for this reason - the theoretical inference guarantees - that both stages follow a linear model.

Including $Rank_{y,\bar{y}}$ as a covariate in Equation 5 serves two important purposes, which Ovaisi et al. [21] do not consider. First, it helps with satisfying the exclusion restriction that Heckman typically requires, namely that there is a variable in the selection equation of the first stage ($Rank_{y,\bar{y}}$) that does not directly impact the outcome variable in the second stage. Violation of the exclusion restriction can lead to severe collinearity between $F_{x,y}$ and $\hat{\boldsymbol{\lambda}}_{x,y}$ in the second stage, resulting in unstable estimates in the second stage. Second, because position bias induces selection bias, the estimation of selection in the first stage is improved by accounting for the various degrees to which an item can be selected based on position.

4 EXPERIMENTS

We evaluate our approach, considering the baselines in Table 1.

4.1 Experimental setup

4.1.1 ***Dataset.*** We conduct experiments using semi-synthetic data based on set 2 from Yahoo! Learning to Rank Challenge [6]. [2] The Yahoo! data contains 1,266 training queries, where each <query, document> pair is represented by a 700-dimensional feature vector as well as expert annotated relevance score ranging from 0 to 4. Following [17, 21], we binarize the relevance scores by considering

[1] We assume that there is no trust bias [2, 32] and, therefore, ranking affects clicks only through observation but ranking does not directly affect clicks as a signal of trust.

[2] https://webscope.sandbox.yahoo.com/catalog.php?datatype=c

Methods	Position bias	Selection bias	Propensity-dependent
Naive SVM^{rank} [15]			No
Propensity SVM^{rank} [17]	✓		Yes
$Heckman^{rank}$ [21]		✓	No
RankAgg [21]	✓	✓	Yes
PIJD	✓	✓	No

Table 1: LTR methods, type of bias each address and whether they are propensity-dependent

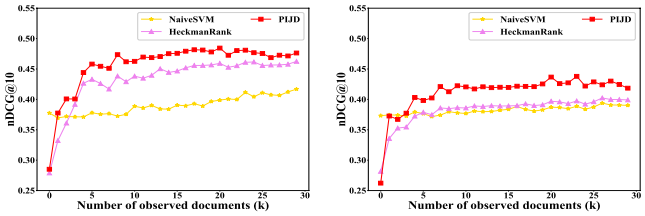
score of 3 and 4 as relevant and others as irrelevant, and randomly sample 70% for training data and 30% for test data. The main reason to choose this data is that it contains the true relevance scores, which allow us to perform an unbiased evaluation.

4.1.2 ***Simulating clicks under position bias and selection bias.*** To generate semi-synthetic datasets that contain biased clicks, we follow the data-generation process of Ovaisi et al. [21] which is closest to our work. We use 1% of the training dataset to train a base ranker (in our case SVM^{rank}), and then use the trained model to generate base rankings for documents with the remaining 99% of the data. We then generate clicks on the ranked documents by considering $P(C_{x,y} = 1)$ to be $\frac{R(x,y)}{(rank_{y,\bar{y}})^{\eta}}$ for items above cut off k and to be 0 otherwise. This reflects a common user click behavior with position and selection bias. To capture the noisy click behavior of users, we allow 10% of clicks to occur for irrelevant results. Similar to Ovaisi et al. [21], we generate clicks over 15 sampling passes over the entire training data.

4.1.3 ***Evaluation.*** We conduct evaluation under 10% click noise, with low and high position bias levels ($\eta = 0.5, 1$), and do not consider severe position bias ($\eta > 1$) where $Heckman^{rank}$ is known to perform poorly [21]. We compare PIJD performance with i) propensity-independent algorithms (*Naive* SVM^{rank}, $Heckman^{rank}$), and ii) propensity-dependent algorithms (*Propensity* SVM^{rank}, *RankAgg*), where the propensity score is perfectly known or misspecified. For the misspecified case, the models are trained with $\eta = 0.5, 1$ but the propensity is misspecified by $\eta = 0.15$. Considering a misspecified propensity estimation captures a more realistic scenario, as propensity estimation relies on only clicked data, and clicks are heavily sparse. We follow [17] and [21] to train *Propensity* SVM^{rank} and $Heckman^{rank}$, where we concatenate results from these two algorithms to train *RankAgg* [21]. Similar to $Heckman^{rank}$, to train *PIJD*, in the first step we use the probit model (Equation 5), and consider the top k results to be possibly observed ($O_{x,y} = 1$) and items below the cut-off k not to be observed ($O_{x,y} = 0$), but unlike $Heckman^{rank}$, we incorporate $Rank_{y,\bar{y}}$ as a covariate. Given the estimated $\hat{\boldsymbol{\theta}}$ from the first stage, we calculate IMR $\lambda_{x,y}$ where it is used in the second stage OLS model as a feature (Equation 6). Finally, given the trained model, LTR algorithms provide ranking for the test set where we evaluate our ranking based on the true unbiased label. We use nDCG@10 as our metric of interest.

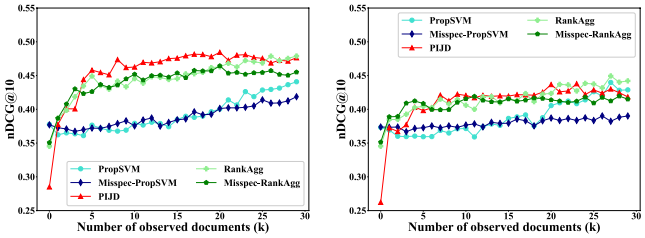
4.2 Results and analysis

4.2.1 ***Comparison with Propensity-independent algorithms.*** Figure 3 displays the performance of propensity-independent LTR methods under various levels of position bias ($\eta \in \{0.5, 1\}$) and selection bias ($k \in [1, 30]$). As depicted, for both $\eta = 0.5$ and $\eta = 1$,



(a) $\eta = 0.5$ **(b)** $\eta = 1$

Figure 3: Propensity-independent LTR algorithms.



(a) True $eta = 0.5$**, assumed** $\eta = 0.15$ **(b) True** $eta = 1$**, assumed** $\eta = 0.15$

Figure 4: Propensity-dependent LTR algorithms with accurate and misspecified propensity.

PIJD outperforms *Naive* SVM^{rank}. This was expected, as *Naive* SVM^{rank} does not account for selection bias nor position bias. Similarly, PIJD outperforms $Heckman^{rank}$. This is likely due to the fact that $Heckman^{rank}$ does not account for position bias nor does it satisfy the exclusion restriction that Heckman requires.

4.2.2 ***Comparison with Propensity-dependent algorithms.*** Figure 4 illustrates the effectiveness of these LTR algorithms, with access to perfect propensity score (*PropSVM*, *RankAgg*) and misspecified propensity score (*Misspec-PropSVM*, *Misspec-RankAgg*). As depicted, *RankAgg* and *PropSVM* outperform their misspecified peers for high cut-off k where the difference is much more pronounced for $\eta = 1$. For $\eta = 0.5$, PIJD clearly outperforms *PropSVM* and its great performance is more noticeable against *Misspec-PropSVM*. This is mainly because *PropSVM* does not account for selection bias. In comparison to *RankAgg*, which accounts for both position and selection bias and is harder to compete with, PIJD outperforms *RankAgg* for $\eta = 0.5$ and the difference is more pronounced against *Misspec-RankAgg*. For $\eta = 1$, similar to $\eta = 0.5$, PIJD clearly outperforms *PropSVM* where the difference is more noticeable against *Misspec-PropSVM*. For $\eta = 1$ PIJD's performance is comparable to *RankAgg* and slightly better than *Misspec-RankAgg*.

5 CONCLUSION

The objective of this paper is to propose a counterfactual framework that corrects for position bias and selection bias in Learning-to-rank systems trained with observational click data without relying on propensity scores. We empirically demonstrate that our proposed approach outperforms LTR algorithms that only account for one bias, and outperforms those that deal with both biases when the position bias is low. Our departure from propensity scores frees our approach from needing online experiments and from estimating biased propensity scores using observational data.

REFERENCES

[1] Aman Agarwal, Kenta Takatsu, Ivan Zaitsev, and Thorsten Joachims. 2019. A General Framework for Counterfactual Learning-to-Rank. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval.

[2] Aman Agarwal, Xuanhui Wang, Cheng Li, Michael Bendersky, and Marc Najork. 2019. Addressing trust bias for unbiased learning-to-rank. In Proceedings of the 28th International Conference on World Wide Web. 4–14.

[3] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. 2019. Estimating position bias without intrusive interventions. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. 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 Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (2018).

[5] Alexey Borisov, Ilya Markov, Maarten de Rijke, and Pavel Serdyukov. 2016. A neural click model for web search. In Proceedings of the 25th International Conference on World Wide Web. 531–541.

[6] Olivier Chapelle and Yi Chang. 2011. Yahoo! learning to rank challenge overview. In Proceedings of the Learning to Rank Challenge. 1–24.

[7] Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. 2012. Large-scale validation and analysis of interleaved search evaluation. In Proceedings of the ACM Transactions on Information Systems (TOIS) (2012).

[8] Olivier Chapelle and Ya Zhang. 2009. A dynamic bayesian network click model for web search ranking. In Proceedings of the 18th international conference on World wide web. ACM, 1–10.

[9] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In Proceedings of the 2008 International Conference on Web Search and Data Mining. ACM, 87–94.

[10] James Heckman. 1979. Sample Selection Bias as a Specification Error. Econometrica 47, 1 (1979), 153–161.

[11] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Probabilistic matrix factorization with non-random missing data. In Proceedings of the 31st International Conference on International Conference on Machine Learning. 1512–1520.

[12] Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten de Rijke. 2013. Reusing historical interaction data for faster online learning to rank for IR. In Proceedings of the sixth ACM International Conference on Web Search and Data Mining. 183–192.

[13] Ziniu Hu, Yang Wang, Qu Peng, and Hang Li. 2019. Unbiased lambdamart: an unbiased pairwise learning-to-rank algorithm. In Proceedings of the 28th International Conference on World Wide Web. 2830–2836.

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

[15] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. 133–142.

[16] Thorsten Joachims, Laura A Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately interpreting clickthrough data as implicit feedback. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

[17] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM, 781–789.

[18] Harrie Oosterhuis and Maarten de Rijke. 2018. Differentiable unbiased online learning to rank. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management. ACM, 1293–1302.

[19] Harrie Oosterhuis and Maarten de Rijke. 2020. Policy-Aware Unbiased Learning to Rank for Top-k Rankings. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (2020).

[20] Harrie Oosterhuis and Maarten de Rijke. 2020. Taking the Counterfactual Online: Efficient and Unbiased Online Evaluation for Ranking. In Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval. 137–144.

[21] Zohreh Ovaisi, Ragib Ahsan, Yifan Zhang, Kathryn Vasilaky, and Elena Zheleva. 2020. Correcting for Selection Bias in Learning-to-rank Systems. In Proceedings of the 29th International Conference on World Wide Web. 1863–1873.

[22] Judea Pearl. 2009. Causality. Cambridge university press.

[23] Karthik Raman and Thorsten Joachims. 2013. Learning socially optimal information systems from egoistic users. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 128–144.

[24] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In Proceedings of the 16th International Conference on World Wide Web.

[25] Yuta Saito. 2020. Doubly Robust Estimator for Ranking Metrics with Post-Click Conversions. In Proceedings of the 14th ACM Conference on Recommender Systems. 92–100.

[26] Yuta Saito. 2020. Unbiased Pairwise Learning from Biased Implicit Feedback. In Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval. 5–12.

[27] Tobias Schnabel and Paul N Bennett. 2020. Debiasing Item-to-Item Recommendations With Small Annotated Datasets. In Proceedings of the 14th ACM Conference on Recommender Systems. 73–81.

[28] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In Proceedings of the International Conference on Machine Learning.

[29] Anne Schuth, Harrie Oosterhuis, Shimon Whiteson, and Maarten de Rijke. 2016. Multileave gradient descent for fast online learning to rank. In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. ACM, 457–466.

[30] Anne Schuth, Floor Sietsma, Shimon Whiteson, Damien Lefortier, and Maarten de Rijke. 2014. Multileaved comparisons for fast online evaluation. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. ACM, 71–80.

[31] Mucun Tian, Chun Guo, Vito Ostuni, and Zhen Zhu. 2020. Counterfactual Learning to Rank using Heterogeneous Treatment Effect Estimation. In Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR eCom) (2020).

[32] 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 & Knowledge Management. 1475–1484.

[33] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to rank with selection bias in personal search. In Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval. 115–124.

[34] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position bias estimation for unbiased learning to rank in personal search. In Proceedings of the 11th ACM International Conference on Web Search and Data Mining.

[35] Yixin Wang, Dawen Liang, Laurent Charlin, and David M Blei. 2020. Causal Inference for Recommender Systems. In Proceedings of the 14th ACM Conference on Recommender Systems.

[36] Bowen Yuan, Jui-Yang Hsia, Meng-Yuan Yang, Hong Zhu, Chih-Yao Chang, Zhenhua Dong, and Chih-Jen Lin. 2019. Improving ad click prediction by considering non-displayed events. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 329–338.

[37] Bowen Yuan, Yaxu Liu, Jui-Yang Hsia, Zhenhua Dong, and Chih-Jen Lin. 2020. Unbiased Ad Click Prediction for Position-aware Advertising Systems. In Proceedings of the 14th ACM Conference on Recommender Systems. 368–377.

[38] Yisong Yue and Thorsten Joachims. 2009. Interactively optimizing information retrieval systems as a dueling bandits problem. In Proceedings of the 26th Annual International Conference on Machine Learning. ACM, 1201–1208.

[39] Ziwei Zhu, Yun He, Yin Zhang, and James Caverlee. 2020. Unbiased Implicit Recommendation and Propensity Estimation via Combinational Joint Learning. In Proceedings of the 14th ACM Conference on Recommender Systems.