Contextual Position Bias Estimation using a Single Stochastic Logging Policy

Giuseppe Di Benedetto[1], Alexander Buchholz[1], Ben London[1], Matej Jakimov[1], Yannik Stein[1], Jan Malte Lichtenberg[1], Vito Bellini[1], Matteo Ruffini[1] and Thorsten Joachims[2,1]

[1]*Amazon*
[2]*Cornell University*

Abstract

Addressing the position bias is of pivotal importance for performing unbiased off-policy training and evaluation in Learning To Rank (LTR). This requires accurate estimates of the probabilities of the users examining the slots where items are displayed, which in many applications is likely to depend on multiple factors, e.g. the screen size. This leads to a position-bias curve that is no longer constant, but depends on the context. Existing position-bias estimators are either non-contextual or require multiple deployed ranking policies. We propose a novel contextual position-bias estimator that only requires propensities logged from a single stochastic logging policy. Empirical evaluations assess the accuracy of the model in recovering the position-bias curves as well as the impact on off-policy evaluation, showing how a contextual position-bias estimator can deliver better reward estimates which are more robust to non-stationarity compared to a non-contextual one.

Keywords

Position-based model, contextual position bias, off-policy evaluation, non-stationarity

1. Introduction

Recommender systems have large catalogs from which to source content to users, and users are usually served with a list of items from which they can choose which items to consume. Optimizing the ranking of presented items heavily impacts the success of recommendation, since users typically only interact with items at the top of a ranking. Industrial systems can leverage vast quantities of past user interactions, which can be used to train new ranking policies and evaluate them offline, before deploying them. Most of the time, these logged interactions only provide implicit feedback that is subject to different sources of biases [1], which need to be addressed both in training and evaluation. For instance, When considering clicks—which arguably constitute the most abundant signal in recommender systems—one cannot directly interpret a non-click as the user not being interested in the recommended item. In fact, when users are presented a list of items to interact with, they can only click on items that the production policy decided to present to the user (i.e., *selection bias*), and they are more likely to examine top positions than bottom ones (i.e., *position bias*) [2]. These biases can be addressed using *click models* that describe how the user interacts with the recommended items [3, 4, 5]. By incorporating these modelling assumptions, we can perform unbiased off-policy training [6, 7, 8] and evaluation [9].

One of the most popular click models is the *position-based model*, which models the click as the realisation of two independent events: examination of the position and relevance of the item (see section 2.1 for more details). To apply this click model to off-policy training and evaluation, one must estimate the vector of examination probabilities for each displayed position, also called the *position-bias curve*. The first methods that appeared in the literature provided estimators for a single position-bias curve to be used for every query [10, 11, 12]. However, in many applications, the examination probabilities are influenced by many factors: the size and shape of the user's screen; the time of day or day of week; the willingness of a user to explore the recommended options; the type of subscription to a paid service, which could limit the number of arbitrary interactions (e.g. number of on-demand streams in a streaming media service) and hence push the user to explore the available options more carefully. One strategy to tackle this dependency consists of partitioning the data and estimating a separate position bias curve for each combination of factors. Unfortunately, this solution would not scale, since (i) the number of combinations grows exponentially with the amount of contextual information, and (ii) for some combinations there might be not enough data for a sufficiently accurate estimate. On the other hand, contextual information can be encoded as features in a parametric model, and recent works [11, 13] have proposed such contextual position-bias estimators to provide examination probabilities at a query level. However, existing models present some limitations, as they either require multiple deployed rankers, or they require accurately estimating the items' relevances, which is arguably as difficult as

LERI 2023 Workshop on Learning and Evaluating Recommendations with Impressions. RecSys'23, September 19, 2023, Singapore

the ranking problem itself.

In this work, we extend the contextual estimator from [14], requiring only a single stochastic policy to be deployed, and for which propensities are known. The contributions of the paper can be summarised as follows:

- We propose *Policy-Aware Contextual Intervention Harvesting* (PA-C-IH), a contextual position-bias estimator, which only requires propensities logged from a single stochastic policy.
- We empirically confirm that the position-bias curve can be accurately recovered when there is dependence on contextual information.
- We explore the impact of contextual position-bias estimation in off-policy evaluation, when using reward estimators relying on the PBM assumption. In particular, we show that contextual position-bias estimation can provide off-policy evaluations that are more accurate and more robust to non-stationarity in the context distribution compared to non-contextual estimation.

2. Background

The process of selecting the best ranking policy to be deployed can be costly and time consuming. Running A/B tests to compare multiple models can negatively affect the user experience, as well as requiring operational effort and time to gather enough data. In addition, A/B testing does not scale when there are many policies to be compared; for example, when considering a large set of hyper-parameter configurations for a neural network-based policy. Off-policy evaluation greatly simplifies this process, allowing comparison of multiple policies using data logged by a previously deployed policy, without the risk of impacting the user experience. However, obtaining accurate off-policy evaluation requires methods to de-bias the estimated rewards. Many estimators have been developed over the past decades [15, 16, 17, 18]. For ranking, these estimators often rely on assumptions about users' click behaviour [19, 9, 7].

2.1. The Position-Based Model

Many off-policy training and evaluation techniques are based on *Inverse Propensity Scoring* (IPS) [20], an importance weighting technique used to counteract biases in the data. In IPS estimators, rewards are re-weighted by the inverse of their probabilities of occurring in the logged data (i.e., propensities). Without any assumptions on users' click behaviour, each of these propensities is the probability that the logging policy produced the *entire ranking*; and due to the combinatorial nature of rankings, this probability could tend to zero, even if the number of items to rank and the number of available slots are not large. Small inverse propensities cause large variance in reward estimates. Hence, assumptions on the users' click behaviour are usually introduced, so as to motivate lower variance estimators. Among the most popular click models, the Position-Based Model (PBM) [21, 2, 19] assumes that clicks on the ranked items are independent, and only characterized by the relevance of the item and the probability of the user examining the position where the item was displayed. Specifically, given a context x, the probability of a click on an item a in position k is

$$\mathbb{P}(C = 1 \mid a, k, x) = \mathbb{P}(E = 1 \mid k, x)\,\mathrm{rel}(a, x)$$

where E denotes the examination random variable, and $\mathrm{rel}(a, x)$ is the relevance of the item a given the context x (i.e. the probability of clicking on that item conditional on having observed it). The object of interest is the examination probability $p_k(x) = \mathbb{P}(E = 1 \mid k, x)$, and for many position-bias estimators, the problem is simplified by assuming there is no dependence of the examination probabilities on the context, reducing the problem to estimating a vector of K—the number of visible slots—probabilities $p = (p_1, \ldots, p_K)$. Contextual position-bias estimation instead focuses on the general case, with the goal of estimating a position-bias curve $p(x)$ for each query defined by a context vector $x \in \mathcal{X}$.

3. Related work

Position-bias estimation plays a central role in developing ranking policies for recommendation and information retrieval, as it provides the weights used to de-bias losses in off-policy training and rewards in off-policy evaluation. Different estimators have been proposed over the years, starting from the simplest approach proposed by Joachims et al. [10], which requires items to be randomly swapped in order to estimate the examination probabilities. Following the PBM assumption, when uniformly swapping items in two positions, k and k', the difference in the CTR logged at those position is due to the difference in the expected examination of the positions; hence, we have $p_k/p_{k'} = \mathrm{CTR}_k/\mathrm{CTR}_{k'}$. Pivoting on a specific position, e.g. the first position, it is possible to consistently estimate the position-bias curve, up to a multiplicative constant, by the CTR ratios using random swaps. These interventions can however be harmful to the user's experience, as displayed items deviate from the optimized policy, pushing non-relevant items in higher positions. Agarwal et al. [11] alleviated this problem by introducing a way to fetch those interventions from multiple different policies deployed online. However, the deployment and maintenance of multiple policies can be cumbersome. Thus Ruffini et al. [12] extended the approach by requiring a single stochastic policy in production. All of the aforementioned works estimate a

single, non-contextual position bias curve, whereas we study contextual position bias estimation. The closest work to ours is by Fang et al. [14], which extends the intervention harvesting approach of Agarwal et al. [11] to contextual position-bias estimation. The downside of this approach is again the requirement of having multiple different policies deployed, which is mitigated by the method proposed in this paper, where we instead use a single stochastic policy with known propensities.

Another stream of research worth mentioning focuses on regression-based estimation. Wang et al. [8] propose estimators that use Expectation-Maximization (EM), and in [22, 13] this method was extended for contextual position-bias estimation. The regression approach has the advantage of not needing randomized data, nor interventions, but at the cost of requiring accurate relevance estimates for the ranked items. The latter requirement is very challenging in practice, and is arguably as hard as solving the ranking problem itself.

4. Contextual position-bias estimator

Like [14], our proposed method does not require explicit interventions, but rather harvests them from already deployed policies. The estimator in [14] requires multiple different policies; each query is served by one of the policies with a pre-defined probability. Here we propose an estimator that instead uses the propensities of a single stochastic logging policy π_0. For each position pair k, k', the *intervention sets* are defined as

$$S_{k,k'} = \{(x, a) : \pi_0(k, a|x)\pi_0(k', a|x) > 0\}$$

and the logging policy π_0 is required to satisfy $S_{k,k'} \neq \emptyset$ for all position pairs $k \neq k'$. This assumption boils down to requiring that for every context x and every pair of positions k, k', there exists at least one action that can be displayed in both positions by the logging policy. This differs from [14] where the intervention sets consisted of items that could have been placed in both positions under the multiple logging policies. As in the case of explicit interventions, the CTRs in the intervention sets can be used to estimate position-bias, with the caveat that in this case the position-bias depends on the context. For each observation in the set of the n click logs $D = \{(x^\ell, c^\ell, a^\ell, k^\ell)\}_{\ell=1:n}$ we can define propensity-weighted click labels as follows:

$$\hat{c}^\ell_{k,k'}(k) := \mathbb{1}_{\{(x^\ell,a^\ell)\in S_{k,k'}\}}\mathbb{1}_{\{k^\ell=k\}}\frac{c^\ell}{\pi_0(k^\ell, a^\ell|x^\ell)}$$

$$\neg\hat{c}^\ell_{k,k'}(k) := \mathbb{1}_{\{(x^\ell,a^\ell)\in S_{k,k'}\}}\mathbb{1}_{\{k^\ell=k\}}\frac{1-c^\ell}{\pi_0(k^\ell, a^\ell|x^\ell)}.$$

Conditioned on the context x, in expectation $\hat{c}^\ell_{k,k'}(k)$ is proportional to the examination probability $p_k(x)$ times the average relevance of the intervention set $r_{k,k'}(x)$. The latter two quantities are hence modelled by two neural networks $h(k, x)$ and $g(k, k', x)$ respectively. It is worth noting that $g(k, k', x)$ aims at estimating the average relevance of the items that can be appear in positions k and k' under the context x, rather than trying to regress on the relevance of each item. The two neural networks can be optimized by minimizing the loss

$$\mathcal{L}(h, g, D) = \sum_{\ell\in D}\sum_{k\neq k'} \hat{c}^\ell_{k,k'}(k)\log\Big(h(k, x^\ell)g(k, k', x^\ell)\Big) + \neg\hat{c}^\ell_{k,k'}(k)\log\Big(1 - h(k, x^\ell)g(k, k', x^\ell)\Big).$$

The contextual position-bias estimator PA-C-IH is thus $\hat{p}(x)_k = h^*(x, k) = \arg\max \mathcal{L}(h, g, D)$. Following analogous steps of Proposition 1 in [14], it can be proven that the loss $\mathcal{L}$ is equivalent to a weighted cross-entropy loss:

$$\sum_{x\in\mathcal{X}}\sum_{k\neq k'} \hat{N}_{k,k'}(x)\Big[\hat{y}_{k,k'}(k, x)\log\Big(h(k, x)g(k, k', x)\Big) + \neg\hat{y}_{k,k'}(k, x)\log\Big(1 - h(k, x)g(k, k', x)\Big)$$

$$\text{where}\quad \hat{N}_{k,k'}(x) := \sum_{\ell\in D}\mathbb{1}_{\{x^\ell=x\}}\mathbb{1}_{\{(x^\ell,a^\ell)\in S_{k,k'}\}}$$

$$\hat{y}_{k,k'}(k, x) = \frac{\sum_{\ell\in D}\mathbb{1}_{\{x^\ell=x\}}\hat{c}^\ell_{k,k'}(k)}{N_{k,k'}(x)}$$

$$\neg\hat{y}_{k,k'}(k, x) = \frac{\sum_{\ell\in D}\mathbb{1}_{\{x^\ell=x\}}\neg\hat{c}^\ell_{k,k'}(k)}{N_{k,k'}(x)}$$

for which $\mathbb{E}[\hat{y}_{k,k'}(k, x)] = h(k, x)g(k, k', x)$ and $\mathbb{E}[\neg\hat{y}_{k,k'}(k, x)] = 1 - h(k, x)g(k, k', x)$ hold. Analogous to [14], in our experiments, both neural networks $h(k, x)$ and $g(k, k', x)$ have one hidden layer with sigmoid activation function in order to force the output to be in the unit interval. The average relevance network $g(k, k', x)$ has an additional hidden layer to ensure that the output is a symmetric matrix; namely, $g(k, k', x) = \frac{1}{2}(g_1(k, k', x)^T + g_1(k, k', x))$, where g_1 denotes the output of the first layer of the network.

5. Experiments

In this section, we empirically compare our contextual estimator PA-C-IH estimator against its non-contextual counterpart, PA-IH [12]. We use synthetic data consisting of 200K queries, with 5 items to be ranked, of which two are relevant. Each query is described by a context vector $x \in \mathbb{R}^5$ sampled from a mixture of three Gaussian distributions $\mathcal{N}(\mu_j, 0.1)$ with cluster means $\mu_1 = (0, 1, -1, 0, 0.5)$, $\mu_2 = (1, 0.2, -0.2, 0.2, 1)$, $\mu_3 = (0.2, 0, 1, 0.3, -0.4)$, and mixture weights $(u_1, u_2, u_3) = (0.3, 0.3, 0.4)$. Following the experimental setup in [14, 13], the examination probabilities for position k, given context x, are defined

as $\mathbb{P}(E = 1 \mid k, x) = \frac{1}{k^{\max(0,\langle\omega,x\rangle+1)}}$. The parameter $\omega \in \mathbb{R}^5$ determines the dependence of the examination probability on the context. Its entries are sampled from Uniform$(-0.5, 0.5)$, and are then fixed for all queries. The logging policy is a deterministic policy selecting the same ranking for all queries, and perturbed by random swaps such that each item maintains its original rank with probability 0.55, or with probability 0.45 is swapped uniformly at random with one of the other items. Clicks are generated according to the contextual PBM.

5.1. Position-bias curve estimation

In order to estimate the position-bias curve, we first tune the hyper-parameters of the two estimators: the optimization parameters for PA-IH and PA-C-IH, and the number of hidden layers for the two neural networks in PA-C-IH. Figure 1 qualitatively shows that the contextual position-bias estimator is able to recover the position-bias curve in each cluster by using the context information, while the non-contextual estimator only fits a position-bias curve that averages across the clusters' position-bias curves. To quantify the accuracy of the position-bias estimates, we compute the relative error,

$$\text{RelError}(\hat{p}) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{K}\sum_{k=1}^{K}\left|1 - \frac{\hat{p}_k(x_i)}{p_k(x_i)}\right|,$$

where N is the number of queries, K is the number of slots in the displayed ranking, and $p_j(x_i)$ and $\hat{p}_j(x_i)$ are the true and estimated examination probabilities for position j in request i with context x_i, respectively. Since position-bias estimates are used in off-policy training and evaluation as inverse propensity scores, this metric can better quantify—as it uses ratios instead of absolute values—how accuracy in position-bias estimation would affect accuracy of off-policy evaluation. Table 1 shows the relative error of PA-C-IH and PA-IH on the synthetic data, showing that the contextual position-bias estimator can lead to significantly improved accuracy.

	PA-C-IH	PA-IH
RelError	0.0556	0.3434
95% CI	[0.0556, 0.0557]	[0.3427, 0.3443]

Table 1
Relative errors in position-bias curve estimation, and 95% bootstrap confidence intervals, of the PA-C-IH and PA-IH estimators on the synthetic dataset.

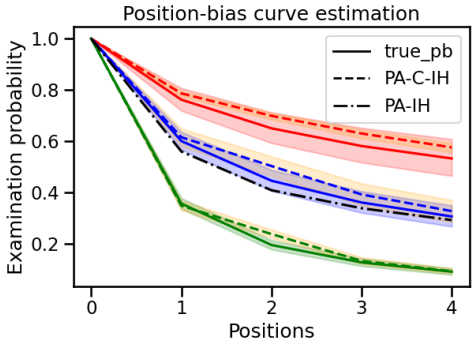

Figure 1: Position-bias estimation in the synthetic data. For each cluster (green, blue, red), the solid line is the average of the true position-bias curves, and the dashed line is the average of the PA-C-IH estimates. Coloured bands are 95% CI of the true position-bias curves (green, blue, red) in the cluster, and of the corresponding PA-C-IH estimates (yellow). The black dash-dotted line is the PA-IH estimate.

5.2. Off-policy evaluation

Among the off-policy estimators developed in the literature (see [9] for a comprehensive overview), an unbiased reward estimator that leverages the PBM assumption in off-policy evaluation is given by

$$\hat{V}(\pi) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} r(a_k)\frac{\langle p(x_i), \pi(\cdot, a_k|x_i)\rangle}{\langle p(x_i), \pi_0(\cdot, a_k|x_i)\rangle}, \quad (1)$$

where π and π_0 are the target and logging policies, respectively; $r(a)$ is the reward logged for item a; $p(x)$ is the position-bias curve for the request with context x, and $\pi(\cdot, a|x)$ denotes a vector of propensities for ranking action a at each of the K positions, given context x. We use this reward estimator below to compare different position-bias estimators for off-policy evaluation.

5.2.1. Stationary environment

In the first experiment, off-policy evaluation was run on the same data source used for position-bias estimation. This setting is realistic under the assumption that the environment does not change over time. Under stationarity, one can expect that a position-bias curve estimated on past data will still be valid when used in the future for off-policy training and evaluation. The target policy to be evaluated here is a deterministic policy that selects among three different rankings, serving the same ranking for all queries within the same cluster. Figure 2 shows the reward estimated on the different clusters, and on the full data set. The PA-C-IH estimator provides much more accurate reward estimates for each of the clusters, as well as the overall data set, compared to PA-IH,

which suffers from the bias introduced by using a single, non-contextual position-bias curve when examination probabilities are in fact contextual.

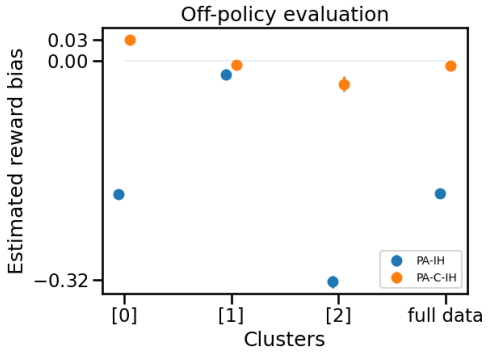

Figure 2: Off-policy evaluation on the synthetic data under stationarity using position-bias estimates from PA-C-IH and PA-IH. Bias of the estimated rewards with 95% CI are reported for each cluster and for the full data.

5.2.2. Non-stationary environment

A less restrictive, and more realistic, setting is where the distribution of queries shifts over time. Position-bias estimation requires data to be collected from a randomized policy, without interventions that can affect the accuracy of the logged propensities (e.g. promotion rules that alter the ranking produced by the policy, thereby invalidating the logged propensities). Such requirements could be difficult to fulfill in real-world applications, thus preventing us from collecting a constant stream of randomized data to update position-bias estimates. In addition to that, it is reasonable to assume that shifts in the context distribution can occur over time, for instance the change in the distribution of the device used, or of the user adhering to different subscription plans, or more generally the non-stationarity induced by the launch of a new user interface. It is therefore interesting to analyze how robust position-bias estimators are under non-stationarity when used for off-policy evaluation. In the synthetic experiment presented, we induce non-stationarity by using a second data set, generated using the same procedure as the data used for position-bias estimation, but with different cluster proportions. While in the training data the cluster weights are $(0.3, 0.3, 0.4)$, in the test data they are set to $(0.15, 0.1, 0.75)$. In order to isolate the effect of non-stationarity in the context distribution, we evaluate a simpler policy than the one used in the previous experiment. Here, the target policy is the deterministic version of the data generation policy—namely, the logging policy without the random swaps used in the data generation step. It is worth recalling that this target policy always selects the same ranking regardless of context. Figure 3 shows the error in the off-policy estimation on the test data, using PA-IH and PA-C-IH position-bias curves estimated on the training data with a different cluster distribution. PA-C-IH proves to be robust to such distribution shifts, providing more accurate position-bias estimates, which translate into more accurate off-policy reward estimates, both within clusters and on the full data set. PA-IH, on the other hand, estimates an overall average position-bias curve, which does not reflect the actual average position-bias curve due to the context distribution shift between the two data sets.

In this experiment the rankings selected by logging and target policies do not depend on the context. Yet even in this very simple scenario, if the position-bias is contextual, a shift in the context distribution can cause systematic bias in off-policy evaluation when using a non-contextual position-bias estimator.

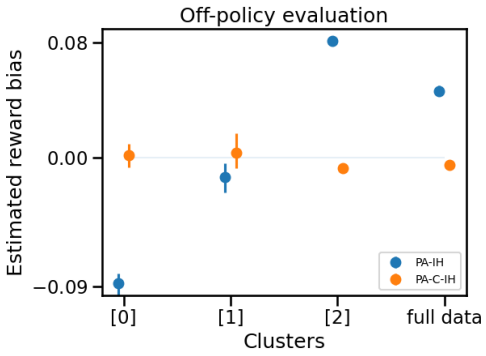

Figure 3: Off-policy evaluation on the synthetic data under non-stationarity using position-bias estimates from PA-C-IH and PA-IH. Bias of the estimated rewards with 95% CI are reported for each cluster and for the full data.

6. Conclusion

We have proposed a new contextual position-bias estimator, PA-C-IH, which does not require multiple rankers to be deployed, but rather a single stochastic ranker for which propensities are known. The latter is commonly adopted in recommender systems in order to ensure a certain level of exploration [23, 24, 25, 12], and our estimator exploits the randomness of the logging policy to provide a contextual estimate of the position-bias curve. We have empirically shown that the PA-C-IH estimator provides better position-bias estimates (compared to a non-contextual estimator) when there is dependence on contextual information, and we explored the impact this

can have on off-policy evaluation. We further demonstrated how PA-C-IH can yield more robust off-policy estimates in the presence of non-stationary distributions.

As part of future work, there are several directions that can be investigated: (i) extend the evaluation of our methods to real-world data; (ii) assess the impact of our estimator in off-policy training of LTR algorithms [26, 6], (iii) generalize our approach to incorporate other types of click noises, such as trust bias.

References

[1] T. Joachims, L. Granka, B. Pan, H. Hembrooke, G. Gay, Accurately interpreting clickthrough data as implicit feedback, in: Acm Sigir Forum, volume 51, Acm New York, NY, USA, 2017, pp. 4–11.

[2] N. Craswell, O. Zoeter, M. Taylor, B. Ramsey, An experimental comparison of click position-bias models, in: Proceedings of the 2008 international conference on web search and data mining, 2008, pp. 87–94.

[3] F. Guo, C. Liu, Y. M. Wang, Efficient multiple-click models in web search, in: Proceedings of the second acm international conference on web search and data mining, 2009, pp. 124–131.

[4] G. E. Dupret, B. Piwowarski, A user browsing model to predict search engine click data from past observations., in: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, 2008, pp. 331–338.

[5] O. Chapelle, Y. Zhang, A dynamic bayesian network click model for web search ranking, in: Proceedings of the 18th international conference on World wide web, 2009, pp. 1–10.

[6] A. Agarwal, K. Takatsu, I. Zaitsev, T. Joachims, A general framework for counterfactual learning-to-rank, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 5–14.

[7] H. Oosterhuis, M. de Rijke, 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, pp. 489–498.

[8] X. Wang, N. Golbandi, M. Bendersky, D. Metzler, M. Najork, Position bias estimation for unbiased learning to rank in personal search, in: Proceedings of the eleventh ACM international conference on web search and data mining, 2018, pp. 610–618.

[9] S. Li, Y. Abbasi-Yadkori, B. Kveton, S. Muthukrishnan, V. Vinay, Z. Wen, Offline evaluation of ranking policies with click models, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, KDD '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 1685–1694. URL: https://doi.org/10.1145/3219819.3220028. doi:10.1145/3219819.3220028.

[10] T. Joachims, A. Swaminathan, T. Schnabel, Unbiased learning-to-rank with biased feedback, in: Proceedings of the tenth ACM international conference on web search and data mining, 2017, pp. 781–789.

[11] A. Agarwal, I. Zaitsev, X. Wang, C. Li, M. Najork, T. Joachims, Estimating position bias without intrusive interventions, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019, pp. 474–482.

[12] M. Ruffini, V. Bellini, A. Buchholz, G. Di Benedetto, Y. Stein, Modeling position bias ranking for streaming media services, in: Companion Proceedings of the Web Conference 2022, 2022, pp. 72–76.

[13] O. B. Mayor, V. Bellini, A. Buchholz, G. Di Benedetto, D. M. Granziol, M. Ruffini, Y. Stein, Ranker-agnostic contextual position bias estimation, arXiv preprint arXiv:2107.13327 (2021).

[14] Z. Fang, A. Agarwal, T. Joachims, Intervention harvesting for context-dependent examination-bias estimation, in: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, 2019, pp. 825–834.

[15] E. L. Ionides, Truncated importance sampling, Journal of Computational and Graphical Statistics 17 (2008) 295–311.

[16] M. Dudík, J. Langford, L. Li, Doubly robust policy evaluation and learning, arXiv preprint arXiv:1103.4601 (2011).

[17] M. Farajtabar, Y. Chow, M. Ghavamzadeh, More robust doubly robust off-policy evaluation, in: International Conference on Machine Learning, PMLR, 2018, pp. 1447–1456.

[18] A. Swaminathan, T. Joachims, The self-normalized estimator for counterfactual learning, advances in neural information processing systems 28 (2015).

[19] A. Chuklin, I. Markov, M. De Rijke, Click models for web search, Springer Nature, 2022.

[20] G. W. Imbens, D. B. Rubin, Causal inference in statistics, social, and biomedical sciences, Cambridge University Press, 2015.

[21] M. Richardson, E. Dominowska, R. Ragno, Predicting clicks: estimating the click-through rate for new ads, in: Proceedings of the 16th international conference on World Wide Web, 2007, pp. 521–530.

[22] Z. Qin, S. J. Chen, D. Metzler, Y. Noh, J. Qin, X. Wang, Attribute-based propensity for unbiased learning in recommender systems: Algorithm and case studies, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge

Discovery & Data Mining, 2020, pp. 2359–2367.
[23] K. Hofmann, S. Whiteson, M. de Rijke, Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval, Information Retrieval 16 (2013) 63–90.
[24] B. Ermis, P. Ernst, Y. Stein, G. Zappella, Learning to rank in the position based model with bandit feedback, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 2405–2412.
[25] J. McInerney, B. Lacker, S. Hansen, K. Higley, H. Bouchard, A. Gruson, R. Mehrotra, Explore, exploit, and explain: personalizing explainable recommendations with bandits, in: Proceedings of the 12th ACM conference on recommender systems, 2018, pp. 31–39.
[26] K. Xiao, X. Cao, P. Huang, S. Chen, X. Zhou, Y. Xian, Learning-to-rank with context-aware position debiasing (2018).