Unbiased Counterfactual Estimation of Ranking Metrics

Haining Yu[4]

[4] Amazon, Seattle WA, USA

Abstract

We propose a novel method to estimate metrics for a ranking policy, based on behavioral signal data (e.g. clicks or viewing of video contents) generated by a second different policy. Building on [1], we prove the counterfactual estimator is unbiased, and discuss its low-variance property. The estimator can be used to evaluate ranking model performance offline, to validate and selection positional bias models, and to serve as learning objectives when training new models.

Keywords

Learning to rank, presentation bias, counterfactual inference

1. Introduction

Ranking algorithms power large scale information retrieval systems. They rank web pages when users look for information in search engines, or products when users shop on e-retailers' websites. Such systems process billions of queries on a daily basis; they also generate large amount of logs. The logs capture online user behavior (e.g., clicking URLs or viewing video contents) and can be used to improve ranking algorithms. As a result, training new ranking models from logs is a central task in Learn To Rank theory and application; it is also often referred to as learning from "implicit feedback" or "counterfactual learn to rank" in literature (e.g., [1]).

Counterfactual learn-to-rank is complicated by the presence of "positional bias". Ranking algorithms determines the position of ranked documents. If the search result page has a "vertical list" layout, the document with rank of 1 is on top of the page; if the result page has a horizontal layout, the document with rank of 1 is on top left corner. When positional bias is present, a document has a higher chance to be examined by user when ranked higher. As a result, when user clicks a document, the click (the "behavioral signal") can be due to one of two reasons: either the document is relevant for the given query, or the document is on top of the list. When positional bias is present, document ranking and relevancy jointly determine behavioral signals, making the signal a noisy proxy for relevancy, the primary goal of ranking optimization.

In the context of counterfactual learn-to-rank, we refer to the algorithm generating the log data as the "behavioral policy". Data generated by behavioral policy is used to train a hopefully better algorithm, called the "target policy". Research work in counterfactual learn-to-rank can be loosely grouped into training and evaluation. For training, the question is how to properly use knowledge in positional bias to train a target policy and maximize relevancy. To start, positional bias models estimate probability for a document to be examined by a user in a given position; the estimation is based on different user behavioral models. Such models, often called "click models", have become widely available; see [2][3][4][5][6][7][8][9][10][11][12]. Built on positional bias models, the seminal work of [1] and [13] established a framework to optimize relevancy using noisy behavioral signal data, proving unbiasedness results for ranking metrics with additive form. For evaluation, the question is how to evaluate the target policy, once trained. For industry ranking applications, the gold standard for evaluation is to A/B test target policy against behavioral policy, collect data on both, and compare ranking metrics such as Average Precision and NDCG. This approach is restricted by limited experimentation time. As an alternative, offline evaluation like [9] predicts target policy ranking metrics using data from behavioral policy.

The research discussed above, in particular [1] and [9], has advanced our understanding to counterfactual learn-to-rank significantly. Meanwhile, each line of research has its pros and cons. Let us use [1] and [9] to highlight. First of all, the two research focuses on different subjects in a causal relationship. Borrowing a causal lens where relevancy and positional bias jointly drive behavioral signals, [1] focuses on relevancy, the "cause" while [9] focuses on behavioral signals, the "effect". It is an open question whether the approach in [1] can be extended to optimize behavioral signal-based metrics (e.g. clicks). Secondly, the two research also differs in validation: once developed, models in [9] can be validated by comparing offline evaluation and online experimentation measurement. For [1], even if we can optimize relevancy, we cannot easily evaluate how much improvement is made, even with online experimentation. This is because evaluating relevancy (and its improvement) ultimately requires manual annotation; for large-scale

---

Causality in Search and Recommendation (CSR) and Simulation of Information Retrieval Evaluation (Sim4IR) workshops at SIGIR, 2021

hainiy@amazon.com (H. Yu)

CEUR Workshop Proceedings (CEUR-WS.org)

online search engines that process billions of queries daily, such effort is costly. Last but not least, [9] relies on high-variance Inverse Propensity Scoring techniques based on the entire ranking permutation (or “action”). In ranking, the action space is large, for example there are $100!/(100-20)! = 1.3 \times 10^{39}$ ways to select 20 documents out of 100. As a result, action probabilities are small. The ratio between two small probabilities can generate extremely small or large ratios (high variance), making the technique challenging to implement in practical situations. Rank-based propensity adjustment in [1] using positional effect models has more desirable variance property. Such difference is key to accuracy of offline evaluation.

This paper brings the two lines of research together. The main contribution is the unbiased estimation for ranking metrics for behavioral signals. In this sense, it is part of study on “effect” of ranking dynamics. By focusing on the “effect”, it can be validated by offline evaluation and experimentation. Meanwhile, it retains the desirable unbiasedness properties [1] and [9], but replaces high variance Inverse Propensity Scoring adjustments with positional biases, borrowing the key insight from [1]. Since the focus switches from cause to effect, this requires new techniques and yields unbiased estimators of a new kind. This unbiased estimator can serve as the learning goal for new target policy and enables offline/online evaluation. It can be also used to establish a method to validate and select positional bias models, a key input to counterfactual estimation framework.

2. Problem Set Up

Let q be a random query. For q, the set of documents to rank is $D = [d_1, d_2, \ldots]$. A ranking policy C assigns ranks $R = [r(d_1), r(d_2), \ldots]$ for documents in D. R, a random permutation of $[1, \ldots, \|D\|]$, determines the position of products D on web page. For example, in a “vertical list” layout, the product with $r = 1$ is on top of the page. After presenting D in order of R to user, we observe the behavioral signal B. A binary vector, $B = [b(d_1), b(d_2), ...]$, where $b(d) = 1$ if and only if user engages with any $d \in D$ (e.g., clicking a web page or watching a video). Given a ranking vector R and the behavioral signal vector B, we define a ranking metric of interest $M = M(R, B)$ such as Precision and DCG.

Table 1 shows a hypothetical example. For $q = 1$, $D = [100, 200, 300]$ represents three documents to rank. The behavioral policy C ranks them as $R^C = [1, 2, 3]$, i.e., to show document 100 first and 300 last. Seeing the list, user ignores the top document 100 and clicks the other two, i.e., $B^C = [0, 1, 1]$. If we use Precision@3 to measure performance of ranking policy, we get $M(R^C, B^C) = 0.667$. Saved in log, the data is used to train target policy T later. The new policy ranks differently, i.e., $R^T = [3, 1, 2]$. This seems an improvement. But user never sees the documents in order of $[200, 300, 100]$ and we don’t know if they will click differently. In other words, B^T is missing data. This is an example of the causal inference, thus the name counterfactual. Estimating $M^T = M(R^T, B^T)$ without observing B^T is the central task for this paper.

Table 1
Illustrative Data Sample

Query	Document	Rank by C	Clicked? (1=yes)	Rank by T
1	100	1	0	3
1	200	2	1	1
1	300	3	1	2

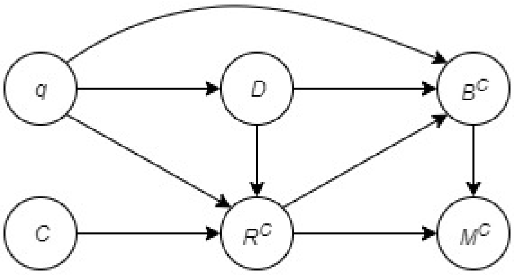

Figure 1: Causal relationship among the random variables

All quantities defined above are random variables (or vectors). The dependency among them are as follows: query q determines the document set D, i.e., $D = D(q)$. q, D, and the ranking policy C jointly determine the rank vector $R = R(q, D, C)$. q, D and R then jointly determine the behavioral signal vector $B = B(q, D, R)$. Last but not least R and B determine $M = M(B, R)$. The table below visualizes the structural causal model.

The randomness in the system comes from multiple sources: distribution of q, conditional distribution of candidate set $D|q$, conditional distribution of $R|q, D, C$, and conditional distribution $B|q, D, R$. The only exception is that no distribution is needed on $M|R, B$. Given R and B, the value of M is deterministic for most practical ranking systems and metrics.

Because the analysis involves two policies, we always specify the policy generating the data. That is, we use R^T to denote ranks generated by policy C, use B^T to denote the behavioral signal generated when showing D in order of R^T to user, and $M^T = M(B^T, R^T)$ to denote the ranking metrics calculated. This helps distinguish between random variables under different policies. We omit other dependencies in notations when confusion can be avoided.

3. The Unbiased Counterfactual Estimator

In this section we define the counterfactual estimator and prove its unbiasedness. We first present the main results in Section 3.1, prove the unbiased results in Section 3.2, and discuss technical details in Section 3.3.

3.1. Main Result

We first define assumptions necessary for defining the estimators and proving unbiasedness. First

Assumption 3.1. *For a policy C, conditional on the ranking vector R^C and behavioral signal B^C, $M^C(R^C, B^C)$ are conditionally independent of q and D, i.e., $M^C(R^C, B^C) \perp\!\!\!\perp q, D | R^C, B^C$*

This is easily satisfied for most ranking metrics such as MRR, MAP, Precision, and NDCG.

Next, similar to [1] [9][13], we assume the ranking metric of interest is linearly decomposable, i.e.,

Assumption 3.2. $M^C(R^C, B^C) = \sum_{d \in D} L(r(d)) b^C(d)$, *where $L(r)$ is a deterministic function of rank r.*

For Precision@3, $L(r) = 1$ if $r \leq 3$, and 0 otherwise.

For $d \in D$, $b^C(d)$ is a binary random variable and $E[b^C(d)]$ is the click probability. Similar to [9] and [10], we make the following assumption based on position-based click model (PBM)[6]:

Assumption 3.3. $E_{B^C|q,D}[b^C(d)] = \eta(r^C(d))\gamma(d, q)$, *where $\eta(r) > 0$ is the probability of examining a certain rank r. $\gamma(d, q)$ is probability of click conditional on being examined.*

When using data generated by behavioral policy C to train a new policy T, we also assume T and C to share nothing in common except inputs q and D. For example, the output of one policy is not used as input to another:

Assumption 3.4. *Given two policies T and C, R^T and R^C are conditionally independent given q and D, i.e., $R^T \perp\!\!\!\perp R^C | q, D$.*

The main result of the paper states that:

Theorem 3.1. *Define*

$$Y(R^C, R^T, B^C) = \sum_{d \in D \text{ and } b^C(d)=1} L(r^T(d)) \frac{\eta(r^T(d))}{\eta(r^C(d))} \tag{1}$$

Let Q_C be queries randomly sampled from the query universe where policy C is applied. Under Assumptions 3.1, 3.2, 3.3, and 3.4,

$$\frac{1}{\|Q_C\|} \sum_{q \in Q_C} Y(R^C, R^T, B^C) \tag{2}$$

is an unbiased estimator for $E_{q,D,R^T,B^T}[M^T(R^T, B^T)]$, i.e.,

$$E_{q,D,R^T,B^T}[M^T(R^T, B^T)] = E\left[\frac{1}{\|Q_C\|} \sum_{q \in Q_C} Y(R^C, R^T, B^C)\right] \tag{3}$$

where the expectation on the right hand side is taken over query set Q_C, D for every $q \in Q_C$,R^T over q, D, B^T over q, D, R^T, and R^C over q, D.

Let us use the same example in Table 1 to illustrate how the estimator is computed. Assume we have the following positional bias estimates: $\eta(1) = 0.9, \eta(2) = 0.7, \eta(3) = 0.5$ (as a reminder, such estimates can be made available via statistical estimation procedures; see [12] and references within for implementation). Recall that ranking vector $R^C = [1, 2, 3]$ and behavioral signal $B^C = [0, 1, 1]$. For the metric of Precision@3, $M^C = 0.667$. For policy T, we observed R^T but not B^T. So we use R^T, R^C, and B^C and equation (1) to compute the following estimate: $Y = (0 + 1 \times \frac{\eta(1)}{\eta(2)} + 1 \times \frac{\eta(2)}{\eta(3)})/3 = 0.895$. Averaging Ys over queries in Q_C yields the counterfactual estimator (2).

3.2. Proof of Unbiasedness

We now set up a series of unbiasedness results, eventually leading to proof of Theorem 3.1.

Lemma 3.2. *Let T and C be two stochastic policies. Under Assumptions 3.1, 3.2, 3.3, and 3.4, $Y(R^C, R^T, B^C)$ is an unbiased estimator for $E_{B^T|q,D,R^T}[M^T]$, i.e.,*

$$E_{B^T|q,D,R^T}[M(R^T, B^T)] = E_{B^C|q,D,R^C,R^T}[Y(R^C, R^T, B^C)] \tag{4}$$

Proof. Via Assumptions 3.1, 3.2 and 3.3

$$E_{B^T|q,D,R^T}[M^T] = \sum_{d \in D} L(r^T(d)) E_{B^T|q,D,R^T}[b^T(d)] \tag{5}$$

By Assumption 3.3,

$$E_{B^T|q,D,R^T}[b^T(d)] = \eta(r^T(d))\gamma(d, q)$$

Defining a shorthand

$$\Psi = \frac{L(r^T(d))}{L(r^C(d))} \frac{\eta(r^T(d))}{\eta(r^C(d))}$$

, it follows that

$$\begin{aligned} & E_{B^T|q,D,R^T}[M^T] \\ = & \sum_{d \in D} L(r^T(d)) \eta(r^T(d)) \gamma(d, q) \\ = & \sum_{d \in D} L(r^C(d)) \eta(r^C(d)) \gamma(d, q) \Psi \end{aligned}$$

$$
\begin{aligned}
&= \sum_{d\in D} L(r^C(d)) E_{B^C|q,D,R^T}[b^C(d)]\Psi \\
&= \sum_{d\in D} L(r^C(d)) E_{B^C|q,D,R^T,R^C}[b^C(d)]\Psi \\
&= E_{B^C|q,D,R^T,R^C}\left[\sum_{d\in D} L(r^C(d)) b^C(d)\Psi\right] \\
&= E_{B^C|q,D,R^T,R^C}\left[\sum_{b^C(d)=1} L(r^T(d))\frac{\eta(r^T(d))}{\eta(r^C(d))}\right] \\
&= E_{B^C|q,D,R^T,R^C}[Y(R^C,R^T,B^C)]
\end{aligned}
$$

The third step in the derivation is due to Assumption 3.3; the fourth step is due to Assumption 3.4. ☐

Lemma 3.3. *Under Assumptions 3.1, 3.2, 3.4, and 3.3,*

$$
\begin{aligned}
& E_{q,D,R^T,B^T}[M(R^T,B^T)] \\
= \; & E_{q,D,R^C,B^C,R^T}[Y(R^C,R^T,B^C)]
\end{aligned}
$$

Proof. By Assumption 3.4, R^T and R^C are conditionally independent. As a result, $M^T(R^T,B^T)$ is also conditionally independent from R^C. Therefore

$$
\begin{aligned}
& E_{B^T|q,D,R^T}[M^T(R^T,B^T)] \\
= \; & E_{B^T|q,D,R^T,R^C}[M^T(R^T,B^T)]
\end{aligned}
$$

Combining this equation with Lemma 3.2 yields

$$
\begin{aligned}
& E_{B^T|q,D,R^T,R^C}[M^T(R^T,B^T)] \\
= \; & E_{B^C|q,D,R^C,R^T}[Y(R^C,R^T,B^C)]
\end{aligned}
$$

The expectations on both sides of the above equation are conditioned on the same joint distribution of q, D, R^T, R^C. Taking expectation over both sides of the equation yields:

$$
\begin{aligned}
& E_{q,D,R^T,R^C,B^T}[M^T(R^T,B^T)] \\
= \; & E_{q,D,R^T,R^C,B^C}[Y(R^C,R^T,B^C)] \qquad (6)
\end{aligned}
$$

Again using Assumption 3.4, we can remove R^C from the expectation on M^T in left hand side of equation (6). This yields

$$
\begin{aligned}
& E_{q,D,R^T,B^T}[M^T(R^T,B^T)] \\
= \; & E_{q,D,R^T,R^C,B^C}[Y(R^C,R^T,B^C)]
\end{aligned}
$$

☐

Theorem 3.1 can now be proved as follows:

Proof. Since $\frac{1}{\|Q_C\|}\sum_{q\in Q_C} Y$ is sample mean of Y, it is an unbiased estimator of the true mean $E_{q,D,R^T,R^C,B^C}[Y]$ which, by Lemma 3.3, equal to $E_{q,D,R^T,B^T}[M^T(R^T,B^T)]$. Thus it is an unbiased estimator of $E_{q,D,R^T,B^T}[M^T(R^T,B^T)]$. ☐

3.3. Technical Discussion

Theorem 3.1 holds when both T and C are deterministic policies, without Assumption 3.4. The proof is omitted due to space limit. In practical ranking systems, output of one ranker is frequently incorporated into another. This violates Assumption 3.4, which requires two policies to share nothing except inputs.

The unbiased estimator looks different from its counterpart in equation (4) of [1], where the positional bias appears only once. It is easy to understand the difference with a causal view: the common assumption in [1] and the current work is that relevancy and positional bias jointly drive behavioral signals. When it comes to estimation, we are interested in different subjects. [1] is interested in estimating relevancy (the cause) from clicks (the effect). So it has the $1/Q$ factor to cancel out the positional bias from behavioral policy. The present work is interested in estimating metrics defined on behavioral signal (the effect) on target policy, from data generated by a behavioral policy policy (a second effect). Two positional bias terms are thus needed to cancel the effect.

The counterfactual estimators (2) aims to avoid the high variance challenge facing other IPS estimators, e.g., in [9]. It is a common practice to use IPS estimators to construct estimates for metrics of interest. While such estimators enjoy the desirable property of unbiasedness, their variance profile is of concern. The core of any IPS estimator is the ratio for a ranking R to be selected by two different policies T and C, i.e., $\Pr^T(R|q,D)/\Pr^C(R|q,D)$. In practice, the ranking space is (combinatorially) large and action probabilities are small. Dividing one small number over another can generate extremely small or large ratios. When any policy is deterministic, $\Pr^T(R|q,D)$ is ill-defined. The problem gets worse when the behavioral and target policy differ significantly, i.e, when accurate offline performance evaluation is needed most. As a result, the ratio can have high variance; this prevents IPS estimators from being useful in industry applications. The current approach solves this problem. Counterfactual estimation using equation (2) no longer needs the high variance action probability ratio. Instead it uses the ratio between positional bias estimates (η), a function of rank positions. The ratio of η empirically has much less variance than estimated action probability ratio.

The current framework can be generalized in three different ways. First, it naturally extends to contextual ranking problems, where q represents not only the search query, but also all context information available for ranker. Secondly, it can be generalized to optimize query/document specific rewards. This makes it easy when different documents have different economic value. Assumption 3.2 can be relaxed to $M(R,B) =$

$\sum_{d \in D} L(r(d), q, D) b(d)$, where $L(r, q, D)$ is a deterministic function of rank r, query q, and document set D. Last but not least, the probability of examination η and condition click probability γ can depend on query q and candidate document set D. That is, Assumption 3.3 can be relaxed to $E_{B^T|q,D}[b^T(d)] = \eta(r^T(d), q, D)\gamma(d, q, D)$. Same is true for Assumption 3.4.

4. Validating and Selecting Positional Bias Models

The unbiased counterfactual estimator has three potential uses: to evaluate offline ranking performance, to validate and selection positional bias estimates, and to serve as loss (or reward in reinforcement learning setting) in training new ranking models. Some have been covered by literature. See [9] for discussion on offline ranking performance evaluation and [1] for discussion on training loss improvement. The rest of this section focuses on validating and selecting positional bias models, an area not covered in past works. positional bias models can be developed in many ways, dependent upon theory (e.g., underlying statistical model, the causal structure, inclusion and exclusion of predictive features), data, and estimation processes. When there is one model, the question is how correct it is. When there are multiple models, the question is how to select the best one for a specific use case.

Using the counterfactual estimator, a method can be developed to validate and selection of positional bias models. It is based on the following idea: we already have one unbiased estimator of $E[M^T]$ using positional bias estimates as input; they are Ys defined in equations (1) and (2). If we find a second unbiased estimators for $E[M^T]$ without using positional bias estimates, the difference between the two estimators can be used to evaluate correctness of positional bias models. Two unbiased estimates of the same quantity (the population mean) should converge. In fact, if we run policy C on a set of queries Q_T, $E[M^T]$ can be directly estimated as $E_{Q_T}\left[\frac{1}{\|Q_T\|}\sum_{q \in Q_T} M(R^T, B^T)\right]$.

The model validation process takes three steps: data collection, estimation, and testing. The first step is to collect data via an online ranking experiment. The experiment should have two treatment groups (C and T), each with a different ranking policy. We then observe behavioral signals (e.g. clicks) for both groups. For every query in T, we also rank the documents with policy C in “shadow mode” and log the ranking from C, even though we don’t know which documents would have been clicked had policy C been applied. The second step is to use the data to compute two unbiased estimators previously defined. In the third step, we use the two estimates to construct a model validation test. A simple approach is to treat the sample mean estimator as the “ground truth”, as long as the sample size of data is big enough. The difference between two estimators can thus be used to quantify the correctness of model. A method with more statistical rigor is to treat the two estimates as group means of random variables with estimated standard deviations. Standard hypothesis testing readily applies.

5. Conclusion

We built a counterfactual estimator for ranking metrics defined on behavioral signals. The estimator is unbiased and has low variance. We discuss its usage in selecting and validating positional bias models. This estimator can be applied to ranking models with strong counterfactual effect.

References

[1] 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.

[2] T. Joachims, Optimizing search engines using clickthrough data, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’02, 2002, p. 133–142.

[3] T. Joachims, L. A. Granka, B. Pan, H. A. Hembrooke, F. Radlinski, G. Gay, Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search, ACM Transactions on Information Systems 25 (2007).

[4] 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, WSDM ’08, 2008, p. 87–94.

[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, WWW ’09, 2009, p. 1–10.

[6] A. Chuklin, I. Markov, M. d. Rijke, Click models for web search, Synthesis Lectures on Information Concepts, Retrieval, and Services 7 (2015) 1–115.

[7] A. Borisov, I. Markov, M. de Rijke, P. Serdyukov, A neural click model for web search, in: Proceedings of the 25th International Conference on World Wide Web, WWW ’16, 2016, p. 531–541.

[8] T. Schnabel, A. Swaminathan, P. Frazier, T. Joachims, Unbiased comparative evaluation of ranking functions, 2016. `arXiv:1604.07209`.

[9] S. Li, Y. Abbasi-Yadkori, B. Kveton, S. Muthukrishnan, V. Vishwa, Z. Wen, Offline evaluation of ranking policies with click models, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp. 1685–1694.

[10] 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 11th ACM International Conference on Web Search and Data Mining (WSDM), 2018, pp. 610–618.

[11] A. Agarwal, I. Zaitsev, T. Joachims, Consistent position bias estimation without online interventions for learning-to-rank, 2018. `arXiv:1806.03555`.

[12] A. Agarwal, I. Zaitsev, X. Wang, C. Li, M. Najork, T. Joachims, Estimating position bias without intrusive interventions, in: WSDM '19: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019.

[13] 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.