Taking the Counterfactual Online: Efficient and Unbiased Online Evaluation for Ranking

Harrie Oosterhuis[1], Maarten de Rijke[12]

September 17, 2020

University of Amsterdam[1], Ahold Delhaize[2]
oosterhuis@uva.nl, derijke@uva.nl
https://staff.fnwi.uva.nl/h.r.oosterhuis
https://staff.fnwi.uva.nl/m.derijke

Main Contributions

Main Contributions

The main contributions of this work are:

- The novel **Logging-Policy Optimization Algorithm (LogOpt)**:
  - Optimizes the logging policy to minimize variance of counterfactual estimation.
- **Proof** of **bias in interleaving methods** under rank-based position biased clicks.

Introduction

Online Ranking Evaluation

The ranking evaluation task:

- Given two rankers which has the highest Click-Through-Rate (CTR)?

Online evaluation methods show rankings to users and observe their clicks.
Based on the observed clicks, they estimate:

- The **absolute** CTR difference or/and the **binary** CTR difference

Goal: Desired Properties

We focus on three estimator properties for ranking evaluation:

- **Consistency** - does the estimation **converge** as more data is gathered.
- **Unbiasedness** - is the **estimate equal** to the **true CTR** difference in expectation.
- **Variance** - the **expected difference** between a single estimate and the mean.

The perfect evaluation method produces an estimator that is consistent and unbiased, while having a minimal amount of variance.

Preliminaries

Assumptions about User Behavior

This paper assumes **rank-based position biased click behavior**.

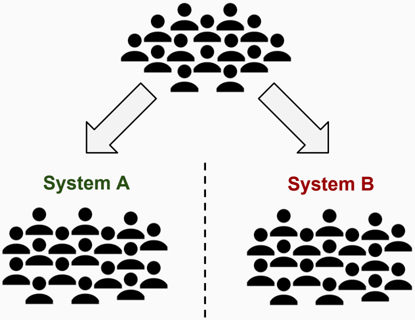
Click probability is a product of an examination probability and a relevance probability. For a document d displayed in ranking R:

$$\underbrace{P(C = 1 \mid R, d)}_{\text{click}} = \underbrace{P(E = 1 \mid \text{rank}(d \mid R))}_{\text{examination}} \underbrace{P(R = 1 \mid d)}_{\text{relevance}}. \quad (1)$$

Existing Evaluation Methods

A/B testing:

- Randomly divide users in two groups, expose each to a different system, observe CTR of each system.
- **Consistent** and **unbiased**.
- Variance depends on the group sizes and actual CTR difference.



Interleaving

Interleaving methods combine rankings of different systems, and infer preferences between them from clicks on the combined rankings.

Methods we considered:

- **Team-Draft Interleaving** - (Radlinski et al., 2008)
- **Probabilistic Interleaving** - (Hofmann et al., 2011)
- **Optimized Interleaving** - (Radlinski and Craswell, 2013)

Interleaving

Interleaving methods combine rankings of different systems, and infer preferences between them from clicks on the combined rankings.

Methods we considered:

- **Team-Draft Interleaving** - (Radlinski et al., 2008)
- **Probabilistic Interleaving** - (Hofmann et al., 2011)
- **Optimized Interleaving** - (Radlinski and Craswell, 2013)

Their properties:

- **Consistent** estimators.
- **Biased** w.r.t. rank-based position bias, we provide a proof by example.
- Variance tested in experiments.

Counterfactual Evaluation

The basis for counterfactual learning to rank is **counterfactual evaluation** using **Inverse Propensity Scoring (IPS) estimators** (Wang et al., 2016; Joachims et al., 2017; Oosterhuis and de Rijke, 2020)

Correct for position bias by **inversely weighting clicks** w.r.t. examination probabilities:

$$P(R = 1 \mid d) = \frac{P(C = 1 \mid R, d)}{P(E = 1 \mid \text{rank}(d \mid R))}. \tag{2}$$

Can be used to **unbiasedly estimate CTR** on ranking R' from clicks on R:

$$P(C = 1 \mid R', d) = \frac{P(C = 1 \mid R, d)}{P(E = 1 \mid \text{rank}(d \mid R))} P(E = 1 \mid \text{rank}(d \mid R')). \tag{3}$$

Counterfactual Evaluation: Policy-Aware Estimator

We use the **Policy-Aware estimator** (Oosterhuis and de Rijke, 2020), which uses the conditional examination probability:

$$P(E = 1 \mid q, d, \underbrace{\pi}_{\text{ranking policy}}) = \sum_R \underbrace{\pi(R \mid q)}_{\text{prob. of } \pi \text{ showing } R} P(E = 1 \mid \text{rank}(d \mid R)). \tag{4}$$

Counterfactual Evaluation: Policy-Aware Estimator

We use the **Policy-Aware estimator** (Oosterhuis and de Rijke, 2020), which uses the conditional examination probability:

$$P(E = 1 \mid q, d, \underbrace{\pi}_{\text{ranking policy}}) = \sum_R \underbrace{\pi(R \mid q)}_{\text{prob. of } \pi \text{ showing } R} P(E = 1 \mid \text{rank}(d \mid R)). \tag{4}$$

When comparing ranking policies π_1 and π_2, using clicks logged with logging policy π_0, a Policy-Aware estimate based on a single query interaction is:

$$f(\pi_0, \pi_1, \pi_2, c, q) = \sum_{d:c(d)=1} \frac{P(E = 1 \mid q, d, \pi_1) - P(E = 1 \mid q, d, \pi_2)}{P(E = 1 \mid q, d, \pi_0)} = \sum_{d:c(d)=1} \frac{\lambda_d}{\rho_d}. \tag{5}$$

Taking the Counterfactual Online

Intuition behind LogOpt

The Policy-Aware counterfactual estimator is **consistent** and **unbiased**.

Variance depends on:

- **The rankers in the comparison.**
- **The users click behavior.**
- **The logging policy used to gather clicks** - we can control this!

Intuition behind LogOpt

The Policy-Aware counterfactual estimator is **consistent** and **unbiased**.

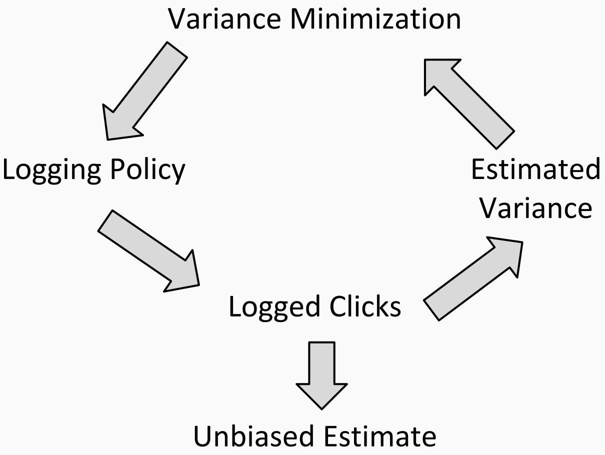
Variance depends on:

- **The rankers in the comparison.**
- **The users click behavior.**
- **The logging policy used to gather clicks** - we can control this!

Our **novel Logging-Policy Optimization Algorithm (LogOpt)** updates the logging policy during the gathering of clicks:

- Turning counterfactual evaluation into online evaluation!

LogOpt Visualization



LogOpt in Detail

LogOpt performs stochastic gradient descent on estimated variance:

$$\overbrace{\text{Var}(\hat{\Delta} \mid q)}^{\text{variance for query } q} = \sum_{c} \overbrace{P(c \mid q)}^{\text{prob. of a click pattern } c} \Bigg(\overbrace{\Delta}^{\text{true CTR diff.}} - \overbrace{\sum_{d:c(d)=1} \frac{\lambda_d}{\rho_d}}^{\text{single estimate}} \Bigg)^2. \tag{6}$$

LogOpt in Detail

LogOpt performs stochastic gradient descent on estimated variance:

$$\overbrace{\text{Var}(\hat{\Delta} \mid q)}^{\text{variance for query } q} = \sum_{c} \overbrace{P(c \mid q)}^{\text{prob. of a click pattern } c} \Bigg(\overbrace{\Delta}^{\text{true CTR diff.}} - \overbrace{\sum_{d:c(d)=1} \frac{\lambda_d}{\rho_d}}^{\text{single estimate}} \Bigg)^2. \tag{6}$$

The derivative reveal two potentially conflicting goals:

$$\begin{aligned}\overbrace{\frac{\delta}{\delta\pi_0}\text{Var}(\hat{\Delta} \mid q)}^{\text{gradient w.r.t. logging policy } \pi_0} &= \sum_{c} \overbrace{\left[\frac{\delta}{\delta\pi_0} P(c \mid q)\right] \left(\Delta - \sum_{d:c(d)=1} \frac{\lambda_d}{\rho_d}\right)^2}^{\text{minimize frequency of high-error click patterns}} \\ &\quad \underbrace{+ P(c \mid q) \left[\frac{\delta}{\delta\pi_0}\left(\Delta - \sum_{d:c(d)=1} \frac{\lambda_d}{\rho_d}\right)^2\right]}_{\text{minimize error of frequent click patterns}}.\end{aligned} \tag{7}$$

LogOpt Solutions

Two problems that LogOpt solves:

- **Problem:** **Relevances** $P(R = 1 \mid d)$ are **unknown** but required for the derivative.
  Solution: Estimate relevances using EM-estimation, following Wang et al. (2016).

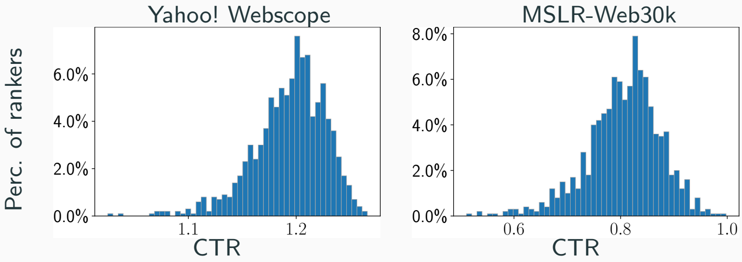
- **Problem:** Derivatives are **computationally infeasible** due to summations over all possible click patterns and all possible rankings.
  Solution: Approximate gradients using Monte-Carlo sampling.

Experiments

Experimental setup

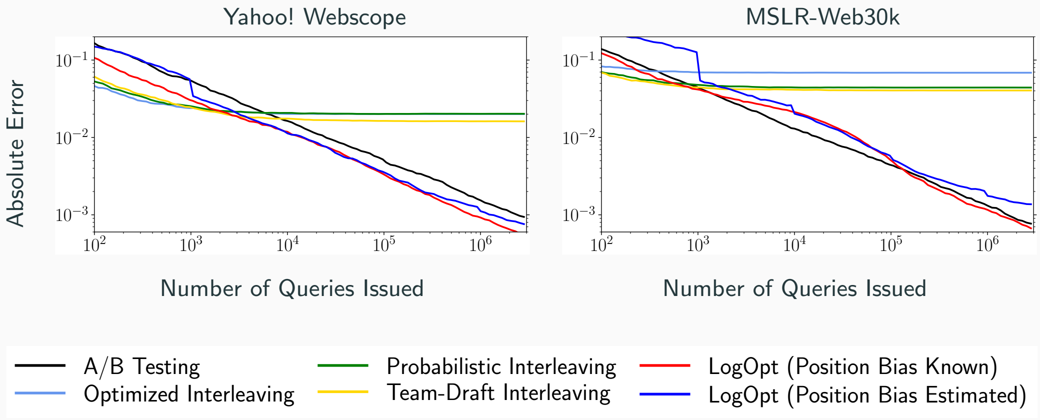
Semi-synthetic experimental setup based on two commercial LTR datasets and simulated position-biased clicks.

Generated 2,000 rankers for 1,000 comparisons, each ranker was trained on a random sample of 100 queries and 50% of features.

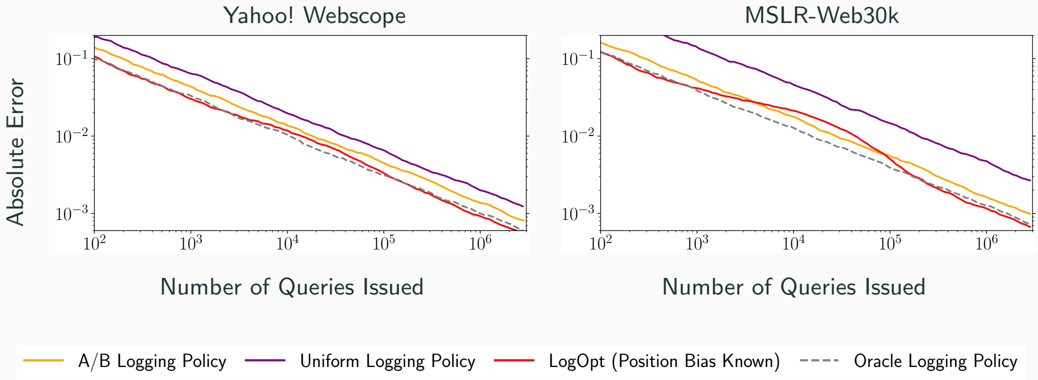


Results

Results: Online Methods - Absolute Error



Results: Logging Policies - Absolute Error



Conclusion

Conclusion

Main takeaways:

- By **optimizing the logging policy**, counterfactual evaluation **turns into online evaluation**.
- We introduced the **Logging-Policy Optimization Algorithm**, our results show that makes counterfactual evaluation **as efficient as online evaluation methods**.

- We proved that **interleaving methods are biased** w.r.t. rank-based position bias, further research needed to understand the impact in practice.

References i

K. Hofmann, S. Whiteson, and M. de Rijke. A probabilistic method for inferring preferences from clicks. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 249–258. ACM, 2011.

T. Joachims, A. Swaminathan, and T. Schnabel. Unbiased learning-to-rank with biased feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 781–789. ACM, 2017.

H. Oosterhuis and 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*. ACM, 2020.

F. Radlinski and N. Craswell. Optimized interleaving for online retrieval evaluation. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 245–254. ACM, 2013.

F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 43–52. ACM, 2008.

X. Wang, M. Bendersky, D. Metzler, and M. Najork. 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*, pages 115–124. ACM, 2016.

Acknowledgments

All content represents the opinion of the author(s), which is not necessarily shared or endorsed by their employers and/or sponsors.