Estimating Clickthrough Bias in the Cascade Model

Praveen Chandar and Ben Carterette*
Spotify
New York City, NY, USA
praveenr,benjaminc@spotify.com

ABSTRACT

Recently, there has been considerable interest in the use of historical logged user interaction data—queries and clicks—for evaluation of search systems in the context of counterfactual analysis [8, 10]. Recent approaches attempt to de-bias the historical log data by conducting randomization experiments and modeling the bias in user behavior. Thus far, the focus has been on addressing bias that arises due to the position of the document being clicked (position-bias) or sparsity of clicks on certain query-document pairs (selection-bias). However, there is another source of bias that could arise: the bias due to the context in which a document was presented to the user. The propensity of the user clicking on a document depends not only on its position but also on many other contextual factors.

In this work, we show that the existing counterfactual estimators fail to capture one type of bias, specifically, the effect on clickthrough rates due to the relevance of documents ranked above. Further, we propose a modification to the existing estimator that takes into account this bias. We rely on full result randomization that allows us to control for the click context at various ranks; we demonstrate the effectiveness of our methods in evaluating retrieval system through experiments on a simulation setup that is designed to cover a wide variety of scenarios.

KEYWORDS

Measurement, Experimentation, Counterfactual Evaluation

ACM Reference Format:
Praveen Chandar and Ben Carterette. 2018. Estimating Clickthrough Bias in the Cascade Model. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3269206.3269315

1 INTRODUCTION

The use of historical log data for evaluation of IR system in an offline setting has been receiving an increasing amount of attention lately [8, 11, 10]. Logs of user interactions with a search system are an extremely valuable resource [6, 3, 4], as they are collected in a natural setting and thus offer a record of behavior untainted by a laboratory setting. Using the logs in an offline evaluation would be helpful for several reasons, including rapid prototyping and testing of retrieval models. However, exploiting historical log data is challenging due to the strong biases present in the logs.

The bias present in logs wasfi rst highlighted through a user study conducted by Joachims et al. [7], in which they investigated the use of clicks for evaluation and training. Their study confirmed the presence of click bias due to the position of a document, i.e. users' clicks are biased towards documents at higher ranks regardless of relevance. Other studies have pointed out different types of biases including presentation bias, attractiveness bias [12], and trust bias [9]. Wang et al. [10] described the challenges that arise due to missing judgments in using historical click data for offline evaluation of new systems.

Existing methods attempt to handle bias in click data with the perspective of *counterfactual analysis*. Wang et al. [10] and Joachims et al. [8] proposed methods that use inverse propensity weighting (IPS) to train and evaluate rankers using historical logs; their focus was on eliminating position bias using some degree of randomization. Since full randomization of results has a high risk of user attrition, Joachims et al. presented a method to minimize the amount of perturbation necessary, so that most users will see most results as intended (we refer to this as the *swap policy*). This was later extended by Wang et al. [11].

These minimally invasive randomization techniques are believed to provide unbiased estimates of effectiveness "value" as long as clicks on documents correlate with relevance. However, work to date has not considered the *context* of a click on a document when it is surrounded by other documents that may be relevant. In particular, for this work, studies on click modeling have shown that the probability of a user reaching a rank is dependent on how satisfied the user was with previously examined documents and several other factors [3, 5]. We refer to this as *cascade bias*, and it is the focus of our work.

There are two main contributions of this work. First, we show through simulations that the existing IPS estimators are biased when the user's behavior follows a cascade-style user browsing model in which their clicks are dependent on the relevance of documents previously examined. Second, we propose a modification to the existing IPS estimator such that it takes into account this cascade bias while estimating the effectiveness score of a ranker. Finally, we demonstrate the effectiveness of the proposed estimator using a simulation setup proposed by Carterette et al. [1].

This paper is organized as follows. In Section 2 we formally describe the counterfactual evaluation task and describe our modified propensity weight-based method. In Section 3 we present our simulation models used in our experiments, and Section 4 presents our experimental results. We conclude in Section 5.

*Work done while the second author was on leave from University of Delaware

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3269315

2 COUNTERFACTUAL EVALUATION

The goal of our evaluation task is to determine the unbiased effectiveness of *any* given alternate ranker given the implicit feedback (e.g. clicks) collected using the production system. Following recent work [8], we use inverse propensity weighting (IPS) and rely on results randomization to remove the bias in the log data.

In this work, we focus on evaluation of rankers that re-rank a fixed set of documents in a candidate pool. We formally introduce our problem and define the adopted notation below:

Let, $\mathcal{L}$ be a set of historical logs and $\ell \in \mathcal{L}$ is a line in the log consisting of the tuple $\langle q_\ell, S_\ell, r_\ell \rangle$; where

- q_ℓ is a query or more generally a *query request* consisting of a query and user profile;
- S_ℓ is a vector of document IDs (the ranked list);
- r_ℓ is the observed reward vector (clicks on documents in ranked list S_ℓ);

The goal of counterfactual evaluation is to determine the unbiased effectiveness of a ranker $\mathcal{S}$ given a set of logs $\mathcal{L}$.

2.1 IPS Estimator

Joachims et al. employ inverse propensity scoring (IPS) to compute the effectiveness of a ranker $\mathcal{S}$ from the logs. The effectiveness score can be computed by iterating over the logs as follows:

$$\hat{V}_{IPS}(\mathcal{S}) = \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} \sum_{d \in S_\ell} \frac{f(\text{rank}(d|\mathcal{S}, q_\ell)) \cdot r_d}{p_r(d|q_\ell)}$$

where,

- $\text{rank}(d|S, q)$ is the rank of document d in the ranked list S (produced by ranker $\mathcal{S}$) for context q;
- $f(\cdot)$ is a function that will be applied to ranks and represents the contribution of the rank of the document to a (linear and additive) IR effectiveness measure such as precision or DCG;
- $p_r(d|q_\ell)$ is the propensity of users to click on the document at rank r for query q_ℓ, which in practice is generalized to the unbiased marginal click-through rate on rank r.

$\hat{V}_{IPS}(\mathcal{S})$ is an unbiased estimator of "value" if we assume that clicks are even somewhat positively correlated with relevance [1], and that are able to estimate propensities without bias.

2.2 Cascade IPS Estimator

While the IPS estimator described above is unbiased, it assumes that the user's browsing behavior depends only on rank position. However, as pointed out by prior studies [5, 3], this assumption is not always true in reality. To explain the implications of this on the IPS estimator, let us consider a simple example. Suppose we have two ranked lists $\mathcal{S}_1$ and $\mathcal{S}_2$ of three documents A, B, C, which are highly relevant, relevant, and non-relevant respectively:

rank	S_1	S_2
1	A	C
2	B	B
3	C	A

[1] We direct the reader to Joachims et al. [8] for the proof

Following the user model described by Chapelle et al. [3], users are more likely abandon the results after clicking on a highly relevant document that completely satisfies their information need. In other words, they are more likely to observe—and click on—B in S_2 than to observe or click on B in S_1. Therefore, the propensity of a click at rank 2 is dependent on the relevance of documents shown at rank 1. The IPS estimator described above does not take into account this cascade bias when estimating the propensities, and would therefore be biased in such scenarios.

To address this problem, we define a measure of propensity that includes the context of documents ranked above, specifically conditioning the probability of a click on a document with the full ranked list as well as the query: $p_r(d|S_\ell, q_\ell)$. In practice, this will again generalize to the probability of a click at a rank, but this time conditioned on the relevance of documents appearing in the ranking above that rank.

$$\hat{V}_{Context-IPS}(\mathcal{S}) = \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} \sum_{d \in S_\ell} \frac{f(\text{rank}(d|\mathcal{S}, q_\ell)) \cdot r_d}{p_r(d|\mathcal{S}_\ell, q_\ell)} \quad (1)$$

2.3 Propensity Model

In order to determine the effectiveness of an alternate ranker using the IPS estimator, we need a way to (1) estimate the propensities p_r, (2) decide on the reward function, and (3) combine the production and randomized logs for value estimation. We briefly describe these components below:

Propensity Estimation – Estimating propensities generally requires unbiased data and cannot be estimated from raw search logs, as there are at least two major sources of bias present [2]: (1) position bias, which means that users are much more likely to click on documents near the top of the ranking, regardless of relevance, than documents ranked lower, and; (2) cascade bias, which means that propensity of a user clicking on a document is conditioned on documents retrieved above it.

To handle these biases in the log data, we rely on two different logging policies for randomizing results:

- *Full-Random Policy* randomizes all documents retrieved by the production ranker such that every document retrieved is shown at every position the same number of times (in expectation). With this data, we can compute unbiased click-through rates for any rank position.
- *Swap Policy* introduced by Joachims et al., is a minimally invasive technique that fixes one rank as an "anchor" and then swaps the document at that rank with another document at a rank picked uniformly at random. This way, every document retrieved will be shown at the anchor rank the same number of times (in expectation). The data collected can then be used to infer the unbiased clickthrough rate for any rank position. We employ this policy to estimate the propensities for the IPS estimator described in Section 2.1.

Reward Function – The reward function in Eq. 1 is intended to discount the rankers for retrieving useful documents at lower ranks. This is similar to the way IR effectiveness metrics are based on functions of the ranks at which relevant documents appear. Joachims

[2] Carterette and Chandar [1] pointed out the bias due to system effectiveness but its irrelevant to our work since we are concerned with only the re-ranking task.

et al. used the sum of the ranks of relevant documents as the reward function in their work. In this work, we follow [1] and define $f(\text{rank})$ in terms of a rank cut-off K as:

$$f_{\text{DCG}}(\text{rank}) = \frac{1}{\log_2(\text{rank}+1)} \text{ if rank} \leq K\text{; 0 otherwise}$$

The expected value of $f(\text{rank})$ is then proportional to DCG@K.

Value Estimation After the swap and insertion policies have been active for some time, we will have two different kinds of logs:

(1) the production ranker logs, which we treat as static such that the ranked results for a given context are identical every time that context appears;
(2) the random logs, collected either using the *Full-Random Policy* or *Swap Policy*. These logs introduce some degree of randomness into logs.

Let $\mathcal{L}, \mathcal{L}_R$ denote these two logs (respectively). The value of the production ranker, or that of any new ranker which simply re-ranks documents ranked by the production ranker $\mathcal{S}_0$, can be evaluated using the contextual IPS estimator as follows:

$$\hat{V}_{Context-IPS}(\mathcal{S}) = \frac{1}{|\mathcal{L} \cup \mathcal{L}_R|} \sum_{\ell \in \mathcal{L} \cup \mathcal{L}_R} \sum_{d \in S_\ell} \frac{f(\text{rank}(d|\mathcal{S}_0, q_\ell)) \cdot r_d}{p_r(d|\mathcal{S}_\ell, q_\ell)}$$

3 EXPERIMENTAL DESIGN

In order to gain a deeper understanding of the counterfactual evaluation methods, we designed our experiments around simulations of queries, rankers, and user interactions. This allowed us to analyze the counterfactual methods under different conditions.

3.1 Simulations

We use a simulation framework similar to the one introduced by Carterette and Chandar [1] that consists of three components which we briefly describe below:

3.1.1 Query Simulation. We generate 1,000 simulated queries, each consisting of a candidate pool set of K documents. We randomly assign relevance to documents in the pool by sampling from a Bernoulli distribution with $p = 0.25$.

3.1.2 Ranker Simulation. Since each ranker in our simulation is ranking the same K documents, they will all have identical precision at rank K. We would like to simulate differences in ranking such that the rankers have different DCG@K.

To simulate a ranker $\mathcal{S}_j$, we first select a parameter η_j uniformly at random from the set $\{2^0, 2^1, 2^2, 2^3, 2^4\}$. This parameter will directly influence the ranker's mean effectiveness. Next, for each query in the simulated pool, we randomly sample a value $\eta_{j,q}$ from a normal distribution with mean η_j and variance proportional to $\sqrt{\eta_j}$. After setting $\eta_{j,q}$, we iteratively sample a relevance grade (uniformly without replacement) and assign it to the document to place at rank k. The relevance values are sampled from a multinomial distribution with probabilities proportional to $\{0 + \eta_{j,q}, 1 + \eta_{j,q}, 2 + \eta_{j,q}, \ldots\}$. Thus the larger $\eta_{j,q}$ is, the "flatter" the multinomial distribution is, creating a greater chance to sample a lower relevance grade at a higher rank. This simulation strategy ensures that no two identical rankers have the exact same value while the choice of η_j ensure that there some rankers that are further apart (and statistically significant).

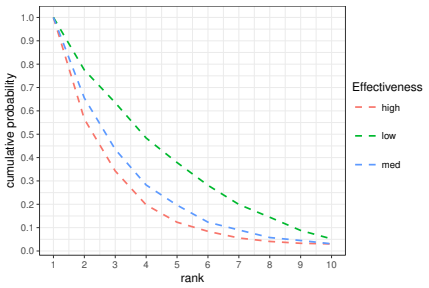

Figure 1: Probability of viewing for three different rankers with *high, medium, low* effectiveness scores.

3.1.3 User Simulation. Our user model is based on the cascade click models introduced by Chappelle et al. [3]. We simulate users proceeding down a ranked list sequentially one result at a time where the chances of stopping are dependent on the previously examined results. More specifically, we use the browsing model of Expected Reciprocal Rank (ERR), where the probability of the user abandoning the ranked list after examining a document at rank k is $s_k = \frac{2^{rel_k}-1}{2^{rel_{max}}}$; where rel_k is the relevance of the document at rank k and rel_{max} is the max relevance grade for any document.

The intuition behind this model is that documents with a higher level of relevance are more likely to satisfy user's information need completely, and thus the follow-up results have a much lower chance of being examined.

Formally, we describe the probability of seeing the next result for a given rank as a function of current rank and relevance of the documents previously examined [3]:

$$P_{ERR}(\text{stopping at rank } k) = \gamma^{k-1} \cdot \prod_{m=1}^{k-1}(1 - s_m)$$

Click Noise: We add click noise to the user simulation to mimic realistic scenarios. The decision of a user to click or not is modeled as binomial conditional on the relevance of the result.

$$P(\text{click at rank } k|S, R_{d_k}) = P(click|R_{d_k}) \cdot S_k$$

where, S_k is probability of viewing rank k given by $1 - P_ERR(k)$, $P(click|R_{d_k} \geq 1) = 0.4$ and $P(click|R_{d_k} = 0) = 0.2$. Figure 1 provides an example with probabilities of viewing for three different rankers with *high, medium, and low* effectiveness. The rankers with *high* effectiveness scores are more likely to rank a *perfect* document[4] at the top compared to rankers with *low* effectiveness.

3.1.4 Simulation Experiment Setup. Finally, we simulate an online testing environment as follows. We randomly chose one of the simulated rankers as the production ranker. The simulation then proceeds in a loop. At each iteration, we simulate a user submitting a query by selecting a one from the query pool at random. Throughout the simulation, there is a 1% chance that the query is diverted to a randomization policy (*swap* or *full-random* and the user sees the perturbed ranked results. The other 99% of the time

[3]Here, we use the extended version of ERR [2], where γ is a persistence parameter and we set it to 1 in our experiments
[4]Typically, a docuemnt with perfect grade is given to the page of a navigational query.

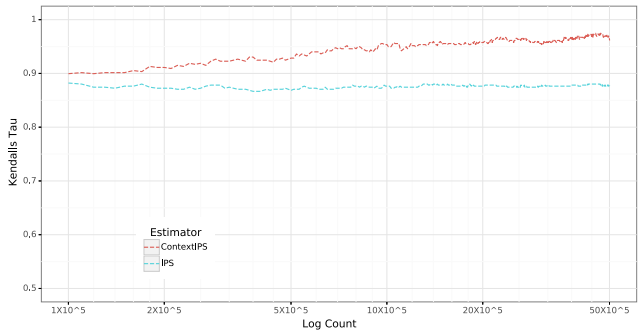

Figure 2: Kendall's tau correlation for the ranking by DCG at $K = 10$ versus the ranking by the $\hat{V}_{\text{IPS}}$ and $\hat{V}_{\text{Context-IPS}}$ estimate of dcg. Each line corresponds to the IPS estimation technique for 1/k user model. Correlation computed every 10,000 log lines.

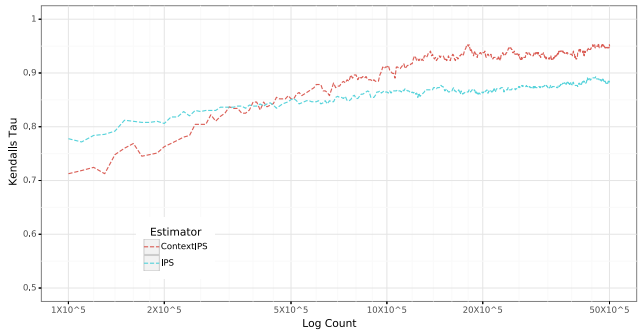

Figure 3: Kendall's tau correlation for the ranking by DCG at $K = 5$ versus the ranking by the $\hat{V}_{\text{IPS}}$ and $\hat{V}_{\text{Context-IPS}}$ estimate of dcg. Each line corresponds to the IPS estimation technique for ERR-Based user model. Correlation computed every 10,000 log lines.

the user will see the original ranked results. For the *swap policy* we use rank 2 for the anchor rank and replace the document from the production ranker with a document selected from a random rank.

4 EXPERIMENTS AND RESULTS

We followed the experiment design described above for simulating a full test environment. We evaluated rankers by DCG@K and used Kendall's tau rank correlation to compare the ranking of rankers by DCG@K to the ranking obtained by the two different IPS estimation methods given in Section 2.1 and Section 2.2. Each experiment was repeated for at least 25 iterations and we report the mean values.

Our primary experimental result demonstrates the ability of the cascade-adjusted IPS method to estimate the click bias when the user follows a cascade-style browsing model. As detailed in Section 3.1.4, we used 1% of the logs to estimate the propensities and evaluate the effectiveness scores of 10 different simulated rankers. Figure 2 shows the tau correlation between the ground truth DCG@K and the score estimated by the IPS methods. The cascade-adjusted IPS exhibits a high correlation of 0.95 once sufficient log lines have been collected, whereas the tau correlations flatten while using the original IPS estimator. This demonstrates that the IPS estimator is insufficient to distinguish between rankers when the user's click is dependent on previously examined documents, while cascade-adjusted IPS is able to identify differences.

Next, we demonstrate the ability of cascade-adjusted IPS to handle presentation bias when it actually does depend only on rank. We replace the ERR-based user model with the one described by Joachims et al. According to this model, the likelihood of a user clicking on a document is a function of its rank alone, i.e. the presentation bias is given by $\frac{1}{rank}$ (see Section 7.1 in [8] for more details). Figure 3 shows that context-IPS exhibits a similar pattern as before with high correlation of 0.95 with the ground truth.

5 CONCLUSION

Detecting and eliminating different sources of bias in historical log data is key to counterfactual evaluation. In this paper, we highlight a specific type of bias that we call *cascade bias* that needs to be handled when using historical logs for offline evaluation. We show that existing IPS estimators are insufficient when the user's browsing behavior follows a cascade-style model, and we propose modifications to handle bias introduced by that behavior. Through experiments on simulated datasets, we show that the proposed estimator has the ability to model such biases. We intend to explore various less invasive logging policies for capturing *contextual bias* and test our methods on real user data in the future.

Acknowledgments This work was supported in part by the National Science Foundation (NSF) under grant number IIS-1350799. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

REFERENCES

[1] B. Carterette and P. Chandar. Offline Comparative Evaluation with Incremental, Minimally-Invasive Online Feedback. *Proc. of SIGIR '18*, 2018.
[2] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 621–630, 2009.
[3] O. Chapelle and Y. Zhang. *A dynamic bayesian network click model for web search ranking*. 2009.
[4] A. Chuklin, I. Markov, and M. de Rijke. Click Models for Web Search. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 7(3):1–115, 2015.
[5] N. Craswell, O. Zoeter, M. J. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. *WSDM*, page 87, 2008.
[6] G. E. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. In *Proceedings of the 31st Annual International ACM*, SIGIR '08, pages 331–338, 2008.
[7] T. Joachims, L. a. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in Web search. *ACM Transactions on Information Systems*, 25(2):7–es, 2007.
[8] 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 - WSDM '17*, 2017.
[9] M. T. Keane and M. O'Brien. Click Models for Web Search. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 28, 2006.
[10] X. Wang, M. Bendersky, D. Metzler, and M. Najork. Learning to Rank with Selection Bias in Personal Search. pages 115–124, 2016.
[11] X. Wang, N. Golbandi, M. Bendersky, D. Metzler, and 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*, WSDM '18, pages 610–618, 2018.
[12] Y. Yue, R. Patel, and H. Roehrig. Beyond position bias: Examining result attractiveness as a source of presentation bias in clickthrough data. In *Proc. of WWW'10*, pages 1011–1018, 2010.