Offline Evaluation of Ranking Policies with Click Models

Shuai Li
The Chinese University of Hong Kong
shuaili@cse.cuhk.edu.hk

Yasin Abbasi-Yadkori
Adobe Research
abbasiya@adobe.com

Branislav Kveton
Adobe Research
kveton@adobe.com

S. Muthukrishnan
Rutgers University
muthu@cs.rutgers.edu

Vishwa Vinay
Adobe Research
vinay@adobe.com

Zheng Wen
Adobe Research
zwen@adobe.com

ABSTRACT

Many web systems rank and present a list of items to users, from recommender systems to search and advertising. An important problem in practice is to evaluate new ranking policies offline and optimize them before they are deployed. We address this problem by proposing evaluation algorithms for estimating the expected number of clicks on ranked lists from historical logged data. The existing algorithms are not guaranteed to be statistically efficient in our problem because the number of recommended lists can grow exponentially with their length. To overcome this challenge, we use models of user interaction with the list of items, the so-called click models, to construct estimators that learn statistically efficiently. We analyze our estimators and prove that they are more efficient than the estimators that do not use the structure of the click model, under the assumption that the click model holds. We evaluate our estimators in a series of experiments on a real-world dataset and show that they consistently outperform prior estimators.

CCS CONCEPTS

• **Information systems** → **Learning to rank**; • **Computing methodologies** → *Learning to rank*; • **Theory of computation** → Online learning algorithms;

KEYWORDS

offline evaluation, ranking, click models, importance sampling

ACM Reference Format:
Shuai Li, Yasin Abbasi-Yadkori, Branislav Kveton, S. Muthukrishnan, Vishwa Vinay, and Zheng Wen. 2018. Offline Evaluation of Ranking Policies with Click Models. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3220028

1 INTRODUCTION

Many web applications, including search, advertising, and recommender systems, generate ranked lists of items and present them to users. Items can be web pages, movies, or products. The industry standard for evaluating the quality of recommended lists is online *A/B testing* [27]. Because A/B testing may impact the experience of users, it is typically used only as the final validation step, while *offline evaluation* is employed in earlier stages [8]. The benefit of offline evaluation is that poor new policies can be identified before they are deployed, and in turn A/B testing can be done more safely and intelligently.

We study the problem of offline evaluation for estimating the expected number of clicks on lists generated by some policy h. This evaluation is done with respect to a previously *logged dataset* S, which records user interactions with lists generated by a *logging production policy* π [21, 22, 28]. Existing algorithms for evaluating h are not guaranteed to be statistically efficient in our setting because they look for exact matches of lists in S. Roughly speaking, since they rely on variants of importance sampling on lists and the number of lists can be exponential in their length, these algorithms may need exponentially many samples from π to perform well.

To overcome this problem of statistical inefficiency, we make structural assumptions on how clicks are generated. In particular, we propose novel unbiased estimators for the expected number of clicks on a list based on well-known click models [7]. The *click model* is a model of user interaction with a ranked list of items, and how the user clicks on these items. Many click models have been proposed, and they represent a spectrum of complexity and accuracy. To illustrate the gain in statistical efficiency from using click models, suppose that all items attract the user independently and that the probability of clicking on an item depends only on its identity. Then it is more efficient to construct a separate estimator for the probability of clicking on each item and then combine these estimators to estimate the expected number of clicks on any given list. Such an estimator would be valid as long as our assumed click model holds, and only requires historical click data.

This paper makes four major contributions:

(1) We formulate the problem of offline evaluation of ranked lists under different click models.
(2) We propose clipped importance sampling estimators for a range of click models, from simple where the clicks are independent of both the item and position, to more realistic where the clicks depend on both the item and its position. All estimators are simple and can be computed with a single pass over historical logged data.
(3) We analyze the properties of our estimators. We show that they have a lower bias than those that ignore the structure of the list, and that the best policy under our estimators has

KDD '18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3220028

a higher lower bound on its value. The analysis is under the assumption that our modeling assumptions hold.

(4) We evaluate our estimators on a large-scale real-world click dataset. A series of experiments shows that our estimators are consistently better than clipped importance sampling at the list level.

We adopt the following notation. Random variables are denoted by boldface letters, lists by uppercase A, and items in the list by lowercase a.

2 SETTING

Let $E = \{1, \ldots, L\}$ be a *ground set* of L *items*, such as web pages. The items are presented to the user in a *list* of length K. Formally, the list is a K*-permutation* of E, which is chosen from set

$$\Pi_K(E) = \{(a_1, \ldots, a_K) : a_1, \ldots, a_K \in E;\ a_i \neq a_j \text{ for any } i \neq j\}.$$

We assume the following general model of user interaction with a list of items. The responses of the user depend on *context* $x \in X$, which is drawn from some distribution over all contexts X. More specifically, let $D(\cdot \mid x)$ be the conditional probability distribution over $\{0, 1\}^{|E| \times K}$ given context x. Then a sample from this distribution, $\boldsymbol{w} \sim D(\cdot \mid x)$, is a matrix where $\boldsymbol{w}(a, k)$ indicates that the user would have *clicked* on item a at position k. The expected value of $\boldsymbol{w}$ given x, $\bar{w}(\cdot \mid x) = \mathbb{E}_{\boldsymbol{w} \sim D(\cdot|x)}[\boldsymbol{w}]$, is a matrix of conditional click probabilities given x. We refer to $\bar{w}(a, k \mid x)$ as the *probability of clicking* on item a at position k given x.

A *policy* π is a conditional probability distribution of a list given context x. We denote this distribution by $\pi(\cdot \mid x)$. The policy interacts with the environment as follows. At time t, the environment draws *context* $\boldsymbol{x}_t$ and *click realizations* $\boldsymbol{w}_t \sim D(\cdot \mid \boldsymbol{x}_t)$. The policy observes $\boldsymbol{x}_t$ and selects list $\boldsymbol{A}_t = (\boldsymbol{a}_1^t, \ldots, \boldsymbol{a}_K^t) \in \prod_K(E)$ according to $\pi(\cdot \mid \boldsymbol{x}_t)$, where $\boldsymbol{a}_k^t$ is the item at position k at time t. Finally, the environment reveals the vector of *item rewards* $(\boldsymbol{w}_t(\boldsymbol{a}_k^t, k))_{k=1}^K$, one entry for each displayed item and its position. The other entries of $\boldsymbol{w}_t$ are unobserved. The *reward* of list $\boldsymbol{A}_t$ is the sum of the observed entries of $\boldsymbol{w}_t$. We define it as $f(\boldsymbol{A}_t, \boldsymbol{w}_t)$, where

$$f(A, w) = \sum_{k=1}^{K} w(a_k, k)$$

for any list $A = (a_1, \ldots, a_K)$ and $w \in [0, 1]^{|E| \times K}$. It follows that the *expected reward* of list A in context x is

$$\mathbb{E}_{\boldsymbol{w} \sim D(\cdot|x)}[f(A, \boldsymbol{w})] = f(A, \bar{w}(\cdot \mid x))\,.$$

Let π be a *logging policy*, which interacts with the environment in n steps and generates a *logged dataset*

$$S = \{(x_t, A_t, w_t)\}_{t=1}^n\,, \tag{1}$$

where $A_t = (a_1^t, \ldots, a_K^t)$ and $w_t(a, k)$ is observed only if $a = a_k^t$. We define the *value of policy* h as

$$\begin{aligned} V(h) &= \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{A} \sim h(\cdot|\boldsymbol{x}),\, \boldsymbol{w} \sim D(\cdot|\boldsymbol{x})}[f(\boldsymbol{A}, \boldsymbol{w})]\right] \\ &= \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{A} \sim h(\cdot|\boldsymbol{x})} f(\boldsymbol{A}, \bar{w}(\cdot \mid \boldsymbol{x}))\right]\,. \end{aligned}$$

Our objective is to estimate $V(h)$ from the logged dataset S in (1). It is common in practice that the logging policy π is unknown and has to be estimated [28]. We denote the *estimated logging policy* by $\hat{\pi}$, and the corresponding conditional probability distribution of a list given context x by $\hat{\pi}(\cdot \mid x)$.

3 ESTIMATORS

In this section, we introduce estimators of $V(h)$ that are motivated by the models of user behavior with a displayed list of items, the so-called click models [7]. We propose multiple estimators, from simple to relatively sophisticated, each mirroring a commonly used click model. Simpler estimators are expected to generalize better, under the assumption that the corresponding click model holds.

3.1 List Estimator

We start with a list-level estimator, which does not leverage the structure of lists and serves as a baseline for our proposed estimators. Recall that $\pi(A \mid x)$ is the probability that policy π chooses list A in response to context x.

Let $h(\cdot \mid x)$ be absolutely continuous with respect to $\pi(\cdot \mid x)$, that is $h(A \mid x) = 0$ when $\pi(A \mid x) = 0$. Then

$$\begin{aligned} V(h) &= \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{A} \sim h(\cdot|\boldsymbol{x}),\, \boldsymbol{w} \sim D(\cdot|\boldsymbol{x})}[f(\boldsymbol{A}, \boldsymbol{w})]\right] \\ &= \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{A} \sim \pi(\cdot|\boldsymbol{x}),\, \boldsymbol{w} \sim D(\cdot|\boldsymbol{x})}\left[f(\boldsymbol{A}, \boldsymbol{w}) \frac{h(\boldsymbol{A} \mid \boldsymbol{x})}{\pi(\boldsymbol{A} \mid \boldsymbol{x})}\right]\right]\,, \end{aligned} \tag{2}$$

where the second equality is from the absolute continuity of $h(\cdot \mid x)$ and $h(A \mid x)/\pi(A \mid x)$ is the *importance weight*. The above change-of-measure trick is known as *importance sampling* [1] and we use it in many forms throughout this paper.

The issue with importance sampling is that $h(A \mid x)/\pi(A \mid x)$ can take large values, which affects the variance of $V(h)$. Therefore, importance sampling estimators are clipped in practice [2, 28]. In this work, we define the *list estimator* as

$$\hat{V}_{\mathrm{L}}(h) = \frac{1}{|S|} \sum_{(x, A, w) \in S} f(A, w) \min\left\{\frac{h(A \mid x)}{\hat{\pi}(A \mid x)}, M\right\}$$

for any estimated logging policy $\hat{\pi}$, logged dataset S, and *clipping constant* $M > 0$. Roughly speaking, the value of policy h on logged dataset S is the sum of clicks on logged lists scaled by their importance weights.

The value of M trades off the bias and variance of the estimator. As $M \to \infty$, the estimator becomes unbiased but its variance may be huge. As $M \to 0$, the variance of the estimator approaches zero because $\hat{V}_{\mathrm{L}}(h) \to 0$ for any logged dataset S.

3.2 Item-Position (IP) Click Model

A popular assumption in click models is that the probability of clicking on item a at position k, $\bar{w}(a, k \mid x)$, depends only on the item and its position [7]. Let

$$\pi(a, k \mid x) = \sum_A \pi(A \mid x) \mathbb{1}\{a_k = a\}$$

be the probability that item a is displayed at position k by policy π in context x. Then a similar importance sampling trick to (2) yields

$$\begin{aligned} V(h) &= \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{A} \sim h(\cdot|\boldsymbol{x}),\, \boldsymbol{w} \sim D(\cdot|\boldsymbol{x})}[f(\boldsymbol{A}, \boldsymbol{w})]\right] \\ &= \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{A} \sim h(\cdot|\boldsymbol{x})}\left[\sum_{k=1}^{K} \bar{w}(\boldsymbol{a}_k, k \mid \boldsymbol{x})\right]\right] \end{aligned} \tag{3}$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\sum_{A} h(A \mid \boldsymbol{x}) \sum_{k=1}^{K} \sum_{a \in E} \bar{w}(a, k \mid \boldsymbol{x}) \mathbb{1}\{a_k = a\}\right]$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\sum_{k=1}^{K} \sum_{a \in E} \bar{w}(a, k \mid \boldsymbol{x}) \sum_{A} h(A \mid \boldsymbol{x}) \mathbb{1}\{a_k = a\}\right] \quad (4)$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\sum_{k=1}^{K} \sum_{a \in E} \bar{w}(a, k \mid \boldsymbol{x})\, h(a, k \mid \boldsymbol{x})\right] \quad (5)$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\sum_{k=1}^{K} \sum_{a \in E} \bar{w}(a, k \mid \boldsymbol{x})\, \pi(a, k \mid \boldsymbol{x}) \frac{h(a, k \mid \boldsymbol{x})}{\pi(a, k \mid \boldsymbol{x})}\right] \quad (6)$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{A} \sim \pi(\cdot \mid \boldsymbol{x}),\, \boldsymbol{w} \sim D(\cdot \mid \boldsymbol{x})}\left[\sum_{k=1}^{K} \boldsymbol{w}(\boldsymbol{a}_k, k) \frac{h(\boldsymbol{a}_k, k \mid \boldsymbol{x})}{\pi(\boldsymbol{a}_k, k \mid \boldsymbol{x})}\right]\right], \quad (7)$$

where (4) is from our assumption that $\bar{w}(a, k \mid \boldsymbol{x})$ only depends on item a and position k; (6) introduces the importance weight; and (7) follows from identities (3) to (5), which are applied in the reverse order to π instead of h.

Following the above derivation, we define the *item-position (IP) estimator* as

$$\hat{V}_{\mathrm{IP}}(h) = \frac{1}{|S|} \sum_{(x, A, w) \in S} \sum_{k=1}^{K} w(a_k, k) \min\left\{\frac{h(a_k, k \mid x)}{\hat{\pi}(a_k, k \mid x)}, M\right\}$$

for any estimated logging policy $\hat{\pi}$, logged dataset S, and clipping constant $M > 0$. Roughly speaking, the value of policy h on logged dataset S is the sum of clicks on logged item-position pairs scaled by their importance weights.

This estimator is expected to be more statistically efficient than $\hat{V}_{\mathrm{L}}$ (Section 3.1) because it depends on fewer importance weights. In particular, the number of the weights in $\hat{V}_{\mathrm{IP}}$ is $O(K|E|)$ and the number of the weights in $\hat{V}_{\mathrm{L}}$ is $O(|\Pi_K(E)|)$. The downside of the IP model, in fact of any model, is that it may not hold.

3.3 Random Click Model (RCM)

The random click model (RCM) is a variant of the IP model (Section 3.2) where the click probability is independent of both the item and its position, that is

$$\bar{w}(a, k \mid x) = \bar{w}(a', k' \mid x)$$

for any a, a', k, k', and x. This model is discussed in Section 3.1 of Chuklin *et al.* [7]. Under this model, $f(A, \bar{w}(\cdot \mid x))$ is independent of A and therefore $V(h) = V(\pi)$ for any policy h. It follows that the value of h can be estimated as

$$\hat{V}_{\mathrm{Random}}(h) = \frac{1}{|S|} \sum_{(x, A, w) \in S} \sum_{k=1}^{K} w(a_k, k)\,,$$

which is the average number of clicks collected by logging policy π. Although the above estimator is simplistic, it is hard to beat in practice when the responses of users do not change significantly with the policy. We return to this issue in Section 6.5.

3.4 Rank-Based Click Model (RCTR)

The rank-based click model (RCTR) is also a variant of the IP model (Section 3.2) where the click probability is independent of the item, that is

$$\bar{w}(a, k \mid x) = \bar{w}(a', k \mid x)$$

for any a, a', k, and x. This model is discussed in Section 3.2 of Chuklin *et al.* [7]. Under this model, the click probability can only depend on the position of the item. However, because all lists are displayed over the same K positions, $f(A, \bar{w}(\cdot \mid x))$ is independent of A and therefore $V(h) = V(\pi)$ for any policy h. It follows that the value of h can be estimated as

$$\hat{V}_{\mathrm{R}}(h) = \frac{1}{|S|} \sum_{(x, A, w) \in S} \sum_{k=1}^{K} w(a_k, k)\,,$$

which is the average number of clicks collected by logging policy π. Note that this is the same estimator as $\hat{V}_{\mathrm{Random}}$ in Section 3.3. Therefore, in the rest of the paper, we only refer to $\hat{V}_{\mathrm{R}}$.

3.5 Position-Based Click Model (PBM)

The position-based click model (PBM) is a variant of the IP model (Section 3.2) where the probability of clicking on item a at position k factors as

$$\bar{w}(a, k \mid x) = \mu(a \mid x)\, p(k \mid x)\,, \quad (8)$$

where $\mu(a \mid x)$ is the conditional probability of clicking on item a in context x given that its position is examined and $p(k \mid x)$ is the *examination probability* of position k in context x. This model was introduced as the *examination hypothesis* in Craswell *et al.* [9] and is discussed in Section 3.3 of Chuklin *et al.* [7]. Joachims *et al.* [18] showed that the examination probability of an item often depends heavily on its position. Note that the PBM is *very different* from the rank-based click model in Section 3.4.

Suppose that the examination probabilities of all positions are known and let $p_x = (p(1 \mid x), \ldots, p(K \mid x))$ be the vector of these probabilities. For any vectors u and v, let $\langle u, v \rangle$ denote their dot product. Then the expected value of any policy h in the PBM can be expressed as

$$V(h) = \mathbb{E}_{\boldsymbol{x}}\left[\sum_{a \in E} \mu(a \mid \boldsymbol{x}) \sum_{k=1}^{K} p(k \mid \boldsymbol{x})\, h(a, k \mid \boldsymbol{x})\right]$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\sum_{a \in E} \mu(a \mid \boldsymbol{x})\, \langle p_{\boldsymbol{x}}, h(a, \cdot \mid \boldsymbol{x}) \rangle \frac{\langle p_{\boldsymbol{x}}, h(a, \cdot \mid \boldsymbol{x}) \rangle}{\langle p_{\boldsymbol{x}}, \pi(a, \cdot \mid \boldsymbol{x}) \rangle}\right]$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{A} \sim \pi(\cdot \mid \boldsymbol{x}),\, \boldsymbol{w} \sim D(\cdot \mid \boldsymbol{x})}\left[\sum_{k=1}^{K} \boldsymbol{w}(\boldsymbol{a}_k, k) \frac{\langle p_{\boldsymbol{x}}, h(\boldsymbol{a}_k, \cdot \mid \boldsymbol{x}) \rangle}{\langle p_{\boldsymbol{x}}, \pi(\boldsymbol{a}_k, \cdot \mid \boldsymbol{x}) \rangle}\right]\right],$$

where the first equality is from identities (3) to (5) in Section 3.2 and our modeling assumption in (8); the second equality introduces the importance weight; and the last equality is from identities (3) to (5), which are applied in the reverse order to π instead of h.

Following the above derivation, we define the *PBM estimator* as

$$\hat{V}_{\mathrm{PBM}}(h) = \frac{1}{|S|} \sum_{(x, A, w) \in S} \sum_{k=1}^{K} w(a_k, k) \min\left\{\frac{\langle p_x, h(a_k, \cdot \mid x) \rangle}{\langle p_x, \hat{\pi}(a_k, \cdot \mid x) \rangle}, M\right\}$$

for any estimated logging policy $\hat{\pi}$, logged dataset S, and clipping constant $M > 0$.

This estimator is expected to be more statistically efficient than $\hat{V}_{\mathrm{IP}}$ (Section 3.2) because it depends on fewer importance weights.

Click model	Assumption
RCM	$\bar{w}(a, k \mid x)$ is independent of both item a and position k
RCTR	$\bar{w}(a, k \mid x)$ only depends on position k
DCTR	$\bar{w}(a, k \mid x)$ only depends on item a
PBM	$\bar{w}(a, k \mid x) = \mu(a \mid x)\, p(k \mid x)$

Table 1: Dependence of click probabilities $\bar{w}(a, k \mid x)$ on item a and its position k in different click models.

In particular, the number of the weights in $\hat{V}_{\text{PBM}}$ is $O(|E|)$ and the number of the weights in $\hat{V}_{\text{IP}}$ is $O(K|E|)$. The downside of the PBM is that it is more restrictive than the IP model.

3.6 Document-Based Click Model (DCTR)

The document-based click model (DCTR), which was introduced as the *baseline hypothesis* in Craswell *et al.* [9], assumes that the probability of clicking on an item depends only on its relevance, that is

$$\bar{w}(a, k \mid x) = \bar{w}(a, k' \mid x)$$

for any a, k, k', and x. Note that this assumption can be viewed as a special case of (8). In particular, it is equivalent to assuming that $p(k \mid x) = 1$ for any k and x; or that $p_x = \mathbf{1}_K$ for any x, where $\mathbf{1}_K$ is a vector of all ones of length K. Therefore, the value of any policy h in the DCTR can be estimated using $\hat{V}_{\text{PBM}}$. In particular, it is

$$\hat{V}_{\text{I}}(h) = \frac{1}{|S|} \sum_{(x,A,w) \in S} \sum_{k=1}^{K} w(a_k, k) \min \left\{ \frac{\langle \mathbf{1}_K, h(a_k, \cdot \mid x) \rangle}{\langle \mathbf{1}_K, \hat{\pi}(a_k, \cdot \mid x) \rangle}, M \right\}$$

for any estimated logging policy $\hat{\pi}$, logged dataset S, and clipping constant $M > 0$. In this work, we refer to this estimator as the *item estimator*.

3.7 Summary

We propose several offline estimators for the average number of clicks on lists of items generated by policy h. The main reason for studying multiple estimators is that the logged dataset S in (1) is finite. When S is small, a simpler estimator may be more accurate because it depends on fewer importance weights, which can be estimated more accurately from less data. On the other hand, when S is large, a more sophisticated estimator may be more accurate because it can capture all nuances of S. This is the so-called *bias-variance tradeoff* and we demonstrate it empirically in Section 6.2. The independence assumptions in our estimators are summarized in Table 1.

We refer to the *item*, *IP*, and *PBM* estimators as being *structured*, because their importance weights depend on individual items in the list. In comparison, the importance weights in the list estimator (Section 3.1) are at the level of the list. Our structured estimators are expected to use logged data more efficiently and we show this empirically in Section 6. We analyze statistical properties of our estimators in the next section.

4 ANALYSIS

In this section, we analyze our estimators from Section 3. Our more general estimators in Section 5 can be analyzed similarly.

This section is organized as follows. In Section 4.1, we show that our structured estimators are unbiased in a larger class of policies than the list estimator. In Section 4.2, we show that our structured estimators estimate the value of any policy with a lower bias than the list estimator. In Section 4.3, we show that the best policy under our structured estimators has a higher value than that under the list estimator. All of our results are derived under the assumption that the corresponding click model holds.

4.1 Unbiased in a Larger Class of Policies

All structured estimators in Section 3 are unbiased in a larger class of policies than the list estimator, under the assumptions that the logging policy is known, $\hat{\pi} = \pi$, and that the corresponding click model holds. We prove this for the IP estimator below.

Proposition 4.1. *Fix any $M > 0$ and a class of policies $\mathcal{H}$. Let*

$$\mathcal{H}_L = \{h \in \mathcal{H} : h(A \mid x)/\pi(A \mid x) \le M \text{ for all } A, x\}$$

be the subset of policies where $\hat{V}_L$ is unbiased, the importance weights are not clipped for any $h \in \mathcal{H}_L$. Let

$$\mathcal{H}_{IP} = \{h \in \mathcal{H} : h(a, k \mid x)/\pi(a, k \mid x) \le M \text{ for all } a, k, x\}$$

be the subset of policies where $\hat{V}_{IP}$ is unbiased, the importance weights are not clipped for any $h \in \mathcal{H}_{IP}$. Then in the IP model (Section 3.2), $\mathcal{H}_L \subseteq \mathcal{H}_{IP}$.

Proof. The proof follows from the observation that

$$\begin{aligned} \frac{h(a, k \mid x)}{\pi(a, k \mid x)} &= \frac{\sum_{A:\, a_k = a} h(A \mid x)}{\sum_{A':\, a'_k = a} \pi(A' \mid x)} \\ &= \sum_{A:\, a_k = a} \frac{h(A \mid x)}{\pi(A \mid x)} \frac{\pi(A \mid x)}{\sum_{A':\, a'_k = a} \pi(A' \mid x)} \\ &\le M \end{aligned}$$

holds for any a, k, and x. □

Similar claims can be derived analogously for both the item and PBM estimators. In particular, let $Y \in \{\text{I}, \text{PBM}\}$ and $\mathcal{H}_Y \subseteq \mathcal{H}$ be the subset of policies where $\hat{V}_Y$ is unbiased, where $\mathcal{H}_Y$ is defined analogously to $\mathcal{H}_{\text{IP}}$. Then $\mathcal{H}_{\text{L}} \subseteq \mathcal{H}_{\text{IP}} \subseteq \mathcal{H}_Y$, under the assumption that the corresponding click model holds. We omit the proofs of these claims due to space constraints.

4.2 Lower Bias in Estimating Policy Values

All structured estimators in Section 3 estimate the value of any policy with a lower bias than the list estimator, under the assumptions that the logging policy is known, $\hat{\pi} = \pi$, and that the corresponding click model holds. We prove this for the IP estimator below.

Proposition 4.2. *Fix any $M > 0$ and policy h. Then in the IP model (Section 3.2), the IP estimator $\hat{V}_{IP}$ has a lower downside bias than the list estimator $\hat{V}_L$,*

$$\mathbb{E}_S[\hat{V}_L(h)] \le \mathbb{E}_S[\hat{V}_{IP}(h)] \le V(h)\,.$$

Proof. The second inequality follows from the observation that the clipping of importance weights leads to a downside bias. The first inequality is proved as follows. First, we note that

$$\mathbb{E}_S[\hat{V}_{\mathrm{L}}(h)] = \mathbb{E}_{\boldsymbol{x}}\left[\sum_{a\in E}\sum_{k=1}^{K}\bar{w}(a,k\mid\boldsymbol{x})\sum_{A:\,a_k=a}\min\{h(A\mid\boldsymbol{x}),M\pi(A\mid\boldsymbol{x})\}\right],$$

$$\mathbb{E}_S[\hat{V}_{\mathrm{IP}}(h)] = \mathbb{E}_{\boldsymbol{x}}\left[\sum_{a\in E}\sum_{k=1}^{K}\bar{w}(a,k\mid\boldsymbol{x})\min\{h(a,k\mid\boldsymbol{x}),M\pi(a,k\mid\boldsymbol{x})\}\right].$$

So, we can prove that $\mathbb{E}_S[\hat{V}_{\mathrm{L}}(h)] \le \mathbb{E}_S[\hat{V}_{\mathrm{IP}}(h)]$ by showing that

$$\sum_{A:\,a_k=a}\min\{h(A\mid x),M\pi(A\mid x)\} \le \min\{h(a,k\mid x),M\pi(a,k\mid x)\}$$

holds for any a, k, and x. The above claim follows from

$$h(a,k\mid x)=\sum_{A:\,a_k=a}h(A\mid x)\ge\sum_{A:\,a_k=a}\min\{h(A\mid x),M\pi(A\mid x)\}$$

and

$$M\pi(a,k\mid x)=\sum_{A:\,a_k=a}M\pi(A\mid x)\ge\sum_{A:\,a_k=a}\min\{h(A\mid x),M\pi(A\mid x)\}\,.$$

This concludes our proof. □

Both the item estimator $\hat{V}_{\mathrm{I}}$ and PBM estimator $\hat{V}_{\mathrm{PBM}}$ are even less biased than the IP estimator $\hat{V}_{\mathrm{IP}}$.

Proposition 4.3. *Fix any $M > 0$ and policy h. Then in the DCTR (Section 3.6),*

$$\mathbb{E}_S[\hat{V}_L(h)] \le \mathbb{E}_S[\hat{V}_{IP}(h)] \le \mathbb{E}_S[\hat{V}_I(h)] \le V(h)\,.$$

Proposition 4.4. *Fix any $M > 0$ and policy h. Then in the PBM (Section 3.5),*

$$\mathbb{E}_S[\hat{V}_L(h)] \le \mathbb{E}_S[\hat{V}_{IP}(h)] \le \mathbb{E}_S[\hat{V}_{PBM}(h)] \le V(h)\,.$$

The above claims can be proved similarly to Proposition 4.2. We omit their proofs due to space constraints.

4.3 Policy Optimization

The estimators in Section 3 can be used to find better production policies. This section provides theoretical guarantees for finding such policies. We start with the list estimator in Section 3.1. Let

$$\tilde{h}_{\mathrm{L}} = \mathrm{argmax}_{h\in\mathcal{H}}\hat{V}_{\mathrm{L}}(h)$$

be the best policy according to the list estimator (Section 3.1). Then the value of $\tilde{h}_{\mathrm{L}}$, $V(\tilde{h}_{\mathrm{L}})$, is bounded from below by the value of the optimal policy as follows.

Theorem 4.5. *Let*

$$h^*_L = \mathrm{argmax}_{h\in\mathcal{H}_L}V(h)$$

be the best policy in the subset of policies $\mathcal{H}_L$, which is defined in Proposition 4.1. Then

$$V(\tilde{h}_L) \ge \tag{9}$$

$$V(h^*_L) - M\mathbb{E}_{\boldsymbol{x}}\left[F_L(\boldsymbol{x}\mid\tilde{h}_L)\right] - M\mathbb{E}_{\boldsymbol{x}}\left[F_L(\boldsymbol{x}\mid h^*_L)\right] - 2K\sqrt{\frac{\ln(4/\delta)}{2|S|}}$$

with probability of at least $1-\delta$, where

$$F_L(x\mid h)=\sum_A\mathbb{1}\left\{\frac{h(A\mid x)}{\pi(A\mid x)}\le M\right\}f(A,\bar{w}(\cdot\mid x))\,\Delta(A\mid x)$$

and $\Delta(A\mid x) = |\hat{\pi}(A\mid x) - \pi(A\mid x)|$ is the error in our estimate of $\pi(A\mid x)$ in context x.

We prove our claim in Section 4.4. Strehl *et al.* [28] proved a similar claim under the assumption that policies are deterministic given context. We generalize this result to stochastic policies.

Now we discuss the bound in (9). It contains three error terms, two expectations over $\boldsymbol{x}$ and one $\sqrt{\log(1/\delta)}$ term. The $\sqrt{\log(1/\delta)}$ term is due to the randomness in generating the logged dataset. The two expectations are due to estimating the logging policy π by $\hat{\pi}$. When the logging policy is known, $\hat{\pi} = \pi$, both terms vanish and our bound reduces to

$$V(\tilde{h}_{\mathrm{L}}) \ge V(h^*_{\mathrm{L}}) - 2K\sqrt{\frac{\ln(4/\delta)}{2|S|}}\,. \tag{10}$$

The $\sqrt{\log(1/\delta)}$ term vanishes as the size of the logged dataset $|S|$ increases.

The best policy under the list estimator, $\tilde{h}_{\mathrm{L}}$, is a solution to the following linear program (LP)

$$\begin{aligned}\max_{h\in\mathcal{H},\,c}\quad & \sum_{(x,A,w)\in S} f(A,w)\,c(A\mid x)\\ \text{s.t.}\quad & c(A\mid x)\,\hat{\pi}(A\mid x)\le h(A\mid x)\,, && \forall A,x\,,\\ & c(A\mid x)\le M\,, && \forall A,x\,,\\ & c(A\mid x)\ge 0\,, && \forall A,x\,,\end{aligned}$$

where c is an auxiliary variable of the same dimension as policy h. The reason is that the maximization of $\min\left\{\frac{h(A\mid x)}{\hat{\pi}(A\mid x)}, M\right\}$ over $h(A\mid x)$ can be equivalently viewed as maximizing $c(A\mid x)$ subject to linear constraints $c(A\mid x)\,\hat{\pi}(A\mid x)\le h(A\mid x)$ and $c(A\mid x)\le M$. Note that although the number of lists A is exponential in K, the above LP has $O(|S|)$ variables.

Similar guarantees can be obtained for all three structured estimators in Section 3. Due to space constraints, we only analyze the value of the best IP policy (Section 3.2).

Theorem 4.6. *Let*

$$h^*_{IP} = \mathrm{argmax}_{h\in\mathcal{H}_{IP}}V(h)\,,\quad \tilde{h}_{IP} = \mathrm{argmax}_{h\in\mathcal{H}}\hat{V}_{IP}(h)\,,$$

where $\mathcal{H}_{IP}$ is defined in Proposition 4.1. Then in the IP model (Section 3.2),

$$V(\tilde{h}_{IP}) \ge \tag{11}$$

$$V(h^*_{IP}) - M\mathbb{E}_{\boldsymbol{x}}\left[F_{IP}(\boldsymbol{x}\mid\tilde{h}_{IP})\right] - M\mathbb{E}_{\boldsymbol{x}}\left[F_{IP}(\boldsymbol{x}\mid h^*_{IP})\right] - 2K\sqrt{\frac{\ln(4/\delta)}{2|S|}}$$

with probability of at least $1-\delta$, where

$$F_{IP}(x\mid h)=\sum_{a\in E}\sum_{k=1}^{K}\mathbb{1}\left\{\frac{h(a,k\mid x)}{\pi(a,k\mid x)}\le M\right\}\bar{w}(a,k\mid x)\,\Delta(a,k\mid x)$$

and $\Delta(a,k\mid x) = |\hat{\pi}(a,k\mid x) - \pi(a,k\mid x)|$ is the error in our estimate of $\pi(a,k\mid x)$ in context x.

The theorem is proved in Section 4.4. Similarly to the list estimator, the expectations over $\boldsymbol{x}$ in (11) vanish when $\hat{\pi} = \pi$, and we get that

$$V(\tilde{h}_{\mathrm{IP}}) \geq V(h^*_{\mathrm{IP}}) - 2K\sqrt{\frac{\ln(4/\delta)}{2|S|}}\,. \tag{12}$$

By Proposition 4.1, $\mathcal{H}_{\mathrm{L}} \subseteq \mathcal{H}_{\mathrm{IP}}$. Therefore, the value of h^*_{IP} is at least as high as that of h^*_{L}, $V(h^*_{\mathrm{IP}}) \geq V(h^*_{\mathrm{L}})$. It follows from (10) and (12) that the lower bound on the value of $\tilde{h}_{\mathrm{IP}}$ is at least as high as that on $\tilde{h}_{\mathrm{L}}$.

The best policy under the IP estimator, $\tilde{h}_{\mathrm{IP}}$, can be computed by solving a similar LP to that for $\tilde{h}_{\mathrm{L}}$. The number of variables in this LP is $O(|S|)$.

4.4 Proofs for Section 4.3

Before we prove Theorem 4.5, we bound the expected value of the list estimator, $\hat{V}_{\mathrm{L}}(h)$, for any policy h.

Lemma 4.7. *Fix the estimated logging policy $\hat{\pi}$ and $M > 0$. Then*

$$\mathbb{E}_S[\hat{V}_L(h)] \leq V(h) + M\mathbb{E}_{\boldsymbol{x}}\left[F_L(\boldsymbol{x} \mid h)\right]\,,$$

$$\mathbb{E}_S[\hat{V}_L(h)] \geq \mathbb{E}_{\boldsymbol{x}}\left[G_L(\boldsymbol{x} \mid h)\right] - M\mathbb{E}_{\boldsymbol{x}}\left[F_L(\boldsymbol{x} \mid h)\right]\,,$$

where $F_L(x \mid h)$ is defined in Theorem 4.5 and

$$G_L(x \mid h) = \sum_A \mathbb{1}\left\{\frac{h(A \mid x)}{\pi(A \mid x)} \leq M\right\} f(A, \bar{w}(\cdot \mid x))\, h(A \mid x)\,.$$

Proof. Note that

$$\begin{aligned}
&\mathbb{E}_S[\hat{V}_{\mathrm{L}}(h)] \\
&\quad = \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{A}\sim\pi(\cdot|\boldsymbol{x})}\left[f(\boldsymbol{A}, \bar{w}(\cdot \mid \boldsymbol{x}))\min\left\{\frac{h(\boldsymbol{A} \mid \boldsymbol{x})}{\hat{\pi}(\boldsymbol{A} \mid \boldsymbol{x})}, M\right\}\right]\right]\,, \\
&\quad = \mathbb{E}_{\boldsymbol{x}}\left[\sum_A\left[f(A, \bar{w}(\cdot \mid \boldsymbol{x}))\min\left\{\frac{h(A \mid \boldsymbol{x})}{\hat{\pi}(A \mid \boldsymbol{x})}, M\right\}\pi(A \mid \boldsymbol{x})\right]\right]\,.
\end{aligned}$$

The main claims are obtained by bounding

$$\min\left\{\frac{h(A \mid \boldsymbol{x})}{\hat{\pi}(A \mid \boldsymbol{x})}, M\right\}\pi(A \mid \boldsymbol{x})$$

from above and below in two cases, when $h(A \mid x)/\pi(A \mid x) \leq M$ (Lemma 4.8) and when $h(A \mid x)/\pi(A \mid x) > M$ (Lemma 4.9). □

Lemma 4.8. *Let $h(A \mid x)/\pi(A \mid x) \leq M$. Then*

$$\min\left\{\frac{h(A \mid x)}{\hat{\pi}(A \mid x)}, M\right\}\pi(A \mid x) \leq h(A \mid x) + M\Delta(A \mid x)\,,$$

$$\min\left\{\frac{h(A \mid x)}{\hat{\pi}(A \mid x)}, M\right\}\pi(A \mid x) \geq h(A \mid x) - M\Delta(A \mid x)\,.$$

Lemma 4.9. *Let $h(A \mid x)/\pi(A \mid x) > M$. Then*

$$0 \leq \min\left\{\frac{h(A \mid x)}{\hat{\pi}(A \mid x)}, M\right\}\pi(A \mid x) \leq h(A \mid x)\,.$$

When the logging policy is known, $\hat{\pi} = \pi$, we have

$$\mathbb{E}_{\boldsymbol{x}}\left[G_{\mathrm{L}}(\boldsymbol{x} \mid h)\right] \leq \mathbb{E}_S[\hat{V}_{\mathrm{L}}(h)] \leq V(h)\,.$$

In expectation, the list estimator underestimates $V(h)$. This is consistent with the intuition that clipping of the estimator leads to a downside bias. Also, when $\hat{\pi} = \pi$, $\mathbb{E}_S[\hat{V}_{\mathrm{L}}(h)] = V(h)$ for all $h \in \mathcal{H}_{\mathrm{L}}$, which means that the list estimator is unbiased for any policy h that is not affected by the clipping.

Now we are ready to prove Theorem 4.5.

Proof. From Hoeffding's inequality and the upper bound in Lemma 4.7,

$$\hat{V}_{\mathrm{L}}(\tilde{h}_{\mathrm{L}}) \leq V(\tilde{h}_{\mathrm{L}}) + M\mathbb{E}_{\boldsymbol{x}}\left[F_{\mathrm{L}}(\boldsymbol{x} \mid \tilde{h}_{\mathrm{L}})\right] + K\sqrt{\frac{\ln(4/\delta)}{2|S|}}$$

with probability of at least $1 - \delta/2$. Similarly, from Hoeffding's inequality, the lower bound in Lemma 4.7, and $\mathbb{E}_{\boldsymbol{x}}\left[G_{\mathrm{L}}(\boldsymbol{x} \mid h^*_{\mathrm{L}})\right] = V(h^*_{\mathrm{L}})$ because $h^*_{\mathrm{L}} \in \mathcal{H}_{\mathrm{L}}$,

$$\hat{V}_{\mathrm{L}}(h^*_{\mathrm{L}}) \geq V(h^*_{\mathrm{L}}) - M\mathbb{E}_{\boldsymbol{x}}\left[F_{\mathrm{L}}(\boldsymbol{x} \mid h^*_{\mathrm{L}})\right] - K\sqrt{\frac{\ln(4/\delta)}{2|S|}}$$

with probability of at least $1 - \delta/2$. The final result follows from the observation that $\hat{V}_{\mathrm{L}}(\tilde{h}_{\mathrm{L}}) \geq \hat{V}_{\mathrm{L}}(h^*_{\mathrm{L}})$. □

Similarly to the above proof, the key step in the proof of Theorem 4.6 are upper and lower bounds on the expected value of the IP estimator, which are presented below.

Lemma 4.10. *Fix the estimated logging policy $\hat{\pi}$ and $M > 0$. Then*

$$\mathbb{E}_S[\hat{V}_{IP}(h)] \leq V(h) + M\mathbb{E}_{\boldsymbol{x}}\left[F_{IP}(\boldsymbol{x} \mid h)\right]\,,$$

$$\mathbb{E}_S[\hat{V}_{IP}(h)] \geq \mathbb{E}_{\boldsymbol{x}}\left[G_{IP}(\boldsymbol{x} \mid h)\right] - M\mathbb{E}_{\boldsymbol{x}}\left[F_{IP}(\boldsymbol{x} \mid h)\right]\,,$$

where $F_{IP}(x \mid h)$ is defined in Theorem 4.6 and

$$G_{IP}(x \mid h) = \sum_{a\in E}\sum_{k=1}^{K}\mathbb{1}\left\{\frac{h(a,k \mid x)}{\pi(a,k \mid x)} \leq M\right\}\bar{w}(a,k \mid x)\, h(a,k \mid x)\,.$$

Proof. The proof follows the same line of reasoning as that in Lemma 4.7, with the exception that we use inequalities

$$\left|\min\left\{\frac{h(a,k \mid x)}{\hat{\pi}(a,k \mid x)}, M\right\}\pi(a,k \mid x) - h(a,k \mid x)\right| \leq M\Delta(a,k \mid x)$$

when $h(a,k \mid x)/\pi(a,k \mid x) \leq M$, and

$$0 \leq \min\left\{\frac{h(a,k \mid x)}{\hat{\pi}(a,k \mid x)}, M\right\}\pi(a,k \mid x) \leq h(a,k \mid x)$$

when $h(a,k \mid x)/\pi(a,k \mid x) > M$. □

Again, when the logging policy is known, $\hat{\pi} = \pi$, we have that

$$\mathbb{E}_{\boldsymbol{x}}[G_{\mathrm{IP}}(\boldsymbol{x} \mid h)] \leq \mathbb{E}_S[\hat{V}_{\mathrm{IP}}(h)] \leq V(h)\,.$$

5 WEIGHTED CLICK ESTIMATORS

Suppose the reward of list A is a weighted sum of clicks,

$$f(A, w) = \sum_{k=1}^{K}\theta_k\, w(a_k, k)$$

for some fixed $\theta = (\theta_1, \ldots, \theta_K) \in \mathbb{R}^K_+$. In Section 3, we study the special case of $\theta = \mathbf{1}_K$. Another important case is when $f(A, w)$ is the *discounted cumulative gain (DCG)* of A, $\theta = \left(\frac{1}{\log_2(1+k)}\right)_{k=1}^{K}$. In this section, we generalize our estimators from Section 3 to any reward function of the above form.

Our generalized estimators are presented in Table 2. These estimators are derived as follows. The IP and RCTR estimators are derived as in Sections 3.2 and 3.4, respectively, with a minor difference that θ_k is carried with $\bar{w}(a,k \mid \boldsymbol{x})$ in all steps of the derivation.

List estimator $\hat{V}_{\mathrm{L}}(h)$
$\frac{1}{\lvert S\rvert}\sum_{(x,A,w)\in S}\sum_{k=1}^{K}\theta_k w(a_k,k)\min\left\{\frac{h(A \mid x)}{\hat{\pi}(A \mid x)}, M\right\}$
IP estimator $\hat{V}_{\mathrm{IP}}(h)$
$\frac{1}{\lvert S\rvert}\sum_{(x,A,w)\in S}\sum_{k=1}^{K}\theta_k w(a_k,k)\min\left\{\frac{h(a_k,k \mid x)}{\hat{\pi}(a_k,k \mid x)}, M\right\}$
RCTR estimator $\hat{V}_{\mathrm{R}}(h)$
$\frac{1}{\lvert S\rvert}\sum_{(x,A,w)\in S}\sum_{k=1}^{K}\theta_k w(a_k,k)$
PBM estimator $\hat{V}_{\mathrm{PBM}}(h)$
$\frac{1}{\lvert S\rvert}\sum_{(x,A,w)\in S}\sum_{k=1}^{K}\theta_k w(a_k,k)\min\left\{\frac{\langle\theta\circ p_x, h(a_k,\cdot \mid x)\rangle}{\langle\theta\circ p_x, \hat{\pi}(a_k,\cdot \mid x)\rangle}, M\right\}$
Item estimator $\hat{V}_{\mathrm{I}}(h)$
$\frac{1}{\lvert S\rvert}\sum_{(x,A,w)\in S}\sum_{k=1}^{K}\theta_k w(a_k,k)\min\left\{\frac{\langle\theta, h(a_k,\cdot \mid x)\rangle}{\langle\theta, \hat{\pi}(a_k,\cdot \mid x)\rangle}, M\right\}$

Table 2: Summary of our estimators. We denote by $u \circ v$ the entry-wise product of vectors u and v.

The PBM estimator is derived as in Section 3.5, with the difference that $\sum_{k=1}^{K}\theta_k\, p(k \mid \boldsymbol{x})\, h(a,k \mid \boldsymbol{x})$ is rewritten as

$$\langle\theta\circ p_{\boldsymbol{x}}, \pi(a,\cdot \mid \boldsymbol{x})\rangle\frac{\langle\theta\circ p_{\boldsymbol{x}}, h(a,\cdot \mid \boldsymbol{x})\rangle}{\langle\theta\circ p_{\boldsymbol{x}}, \pi(a,\cdot \mid \boldsymbol{x})\rangle},$$

where $u \circ v$ is the entry-wise product of vectors u and v. The item estimator is derived from the PBM estimator, as in Section 3.6.

6 EXPERIMENTS

We experiment with the *Yandex* dataset [33], which is a web search dataset with more than 167 million web search queries. The dataset contains a training set, which is recorded over 27 days, and a test set, which is recorded over 3 days. Each *record* in the Yandex dataset contains a query ID, the day when the query occurs, 10 displayed items as a response to the query, and the corresponding click indicators of each displayed item.

We observe in a majority of queries that the average number of clicks in the test set is significantly lower than in the training set, sometimes by an order of magnitude. This is due to the preprocessing of the Yandex dataset for the *Personalized Web Search Challenge* [33]. Due to this downside bias, all structured estimators in Section 3 perform extremely well at $M < 1$, when the training set is used as the logged dataset S in (1) and the test set is used to estimate $V(h)$. To avoid this systematic bias, which does not show the statistical efficiency of our estimators, we discard the test set and adopt a different evaluation methodology.

6.1 Experimental Setup

We compare five estimators from Section 3: list $\hat{V}_{\mathrm{L}}$, RCTR $\hat{V}_{\mathrm{R}}$, item $\hat{V}_{\mathrm{I}}$, IP $\hat{V}_{\mathrm{IP}}$, and PBM $\hat{V}_{\mathrm{PBM}}$. They are implemented as described in Section 3. The examination probability of position k in the PBM is set to $1/k$. We leave its optimization for future work.

For each estimator, query q, and day $d \in [27]$, we put all records in day d into the *evaluation set* and all records in days $[27]\setminus\{d\}$ into the *production set*. The production set is the logged dataset S in (1). We estimate the *production policy* by the frequencies of lists in the production set, and denote it by $\hat{\pi}_{q,d}$. We estimate the *evaluated policy* by the frequencies of lists in the evaluation set, and denote it by $h_{q,d}$. Let $V_{q,d}$ be the value of $h_{q,d}$ on day d in query q, which is estimated by its empirical average in the evaluation set; and $\hat{V}_{q,d}$ be its estimate from $\hat{\pi}_{q,d}$. We measure the error of the estimator in query q by its average error over all evaluation sets, one for each day. In particular, we use the *root-mean-square error (RMSE)* in

$$\sqrt{\tfrac{1}{27}\textstyle\sum_{d=1}^{27}(\hat{V}_{q,d}-V_{q,d})^2}\,.$$

The error in multiple queries is defined as

$$\sqrt{\tfrac{1}{27\lvert Q\rvert}\textstyle\sum_{q\in Q}\sum_{d=1}^{27}(\hat{V}_{q,d}-V_{q,d})^2}\,,$$

where Q is the set of the evaluated queries.

All estimators are evaluated on three prediction problems: the expected number of clicks at the first $K = 2$ positions, where our dataset is restricted to those positions; the expected number of clicks at the first $K = 3$ positions, where our dataset is restricted to those positions; and the DCG, where the estimators are weighted as described in Section 5. Our estimators yield only minor improvements in predicting the expected number of clicks at all positions. We discuss this issue in detail in Section 6.5.

The queries in the Yandex dataset do not come with context. Therefore, we assume that the context is the same in all records.

6.2 Illustrative Query

This experiment illustrates our setup and its variations. It is conducted on query 11655238, which appears in our dataset 3 553 times. The responses are 63 distinct lists and 53 distinct items.

The prediction errors at the first $K = 2$ positions are reported in Figure 1a. The error of the RCTR estimator is 0.075. For any $M \geq 40$, the errors of our three structured estimators are at least 16.07% lower than that of the RCTR estimator and 13.55% lower than that of the list estimator. The error of the list estimator at $M = 5$ is 4.50% lower than that at $M = \infty$. The error of the IP estimator at $M = 5$ is 1.74% lower than that at $M = \infty$. This shows the benefits of clipping in importance sampling estimators.

The prediction errors at the first $K = 3$ positions are reported in Figure 1b. For any $M \geq 30$, the errors of our three structured estimators are at least 13.56% lower than that of the RCTR estimator and 10.51% lower than that of the list estimator. These gains further increase to 19.59% and 51.73% at $M = 5$.

The DCG prediction errors are reported in Figure 1c. The errors of the list estimator never drop below 0.245, and therefore are not visible in the figure. For any $M \geq 100$, the errors of our structured estimators are at least 6.82% lower than that of the RCTR estimator and 68.34% lower than that of the list estimator. These gains further increase to 10.32% and 72.86% at $M = 9$.

Finally, we observe in all figures that the item estimator consistently outperforms the IP estimator. We believe that this is because of the bias-variance tradeoff. In particular, the item estimator has a higher bias than the IP estimator because it is a special case of that estimator (Sections 3.5 and 3.6). Because of that, it depends on

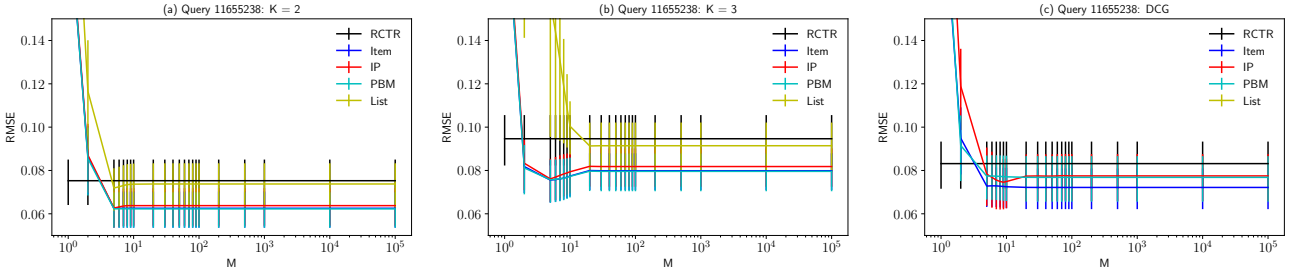

Figure 1: Prediction errors on query 11655238 as a function of clipping parameter M.

fewer estimated importance weights, and can perform better in the regime of less training data, as in this query.

6.3 Hundred Most Frequent Queries

Our second experiment is conducted on 100 most frequent queries. The number of records in these queries ranges from 15k to 455k, and the number of distinct lists ranges from 69 to 10k. This experiment validates our findings from Section 6.2 at a larger scale.

The errors of all estimators are reported in Figure 2. The errors are averaged over all queries and days, as described in Section 6.1. Similarly to Section 6.2, we observe that our structured IP estimator outperforms both of our baselines. For any $M \geq 100$, the error of the IP estimator is at least 17.90% ($K = 2$), 46.24% ($K = 3$), 81.96% (DCG) lower than that of the list estimator. The performance of the list estimator worsens dramatically from $K = 2$ to $K = 3$ because the number of distinct lists over three positions is typically much larger than over two. For any $M \geq 100$, the error of the IP estimator is at least 13.18% ($K = 2$), 12.50% ($K = 3$), 10.65% (DCG) lower than that of the RCTR estimator.

6.4 Less Frequent Queries

Our last experiment is conducted on the tail 900 queries from 1k most frequent queries. These queries are much less frequent than those in Section 6.3, some with as few as 3k records.

The errors of all estimators are reported in Figure 3. We observe that the absolute errors of all estimators increase as we consider less frequent queries. This is expected since less frequent queries provide less training data. Nevertheless, our estimators still improve over baselines. In particular, the IP estimator improves consistently over both the RCTR and list estimators.

6.5 Bias in the Yandex dataset

In our initial experiments, we estimated the expected number of clicks at all $K = 10$ positions. Our estimators performed poorly (Figure 4). More specifically, only the IP estimator improved over the RCTR estimator, and that improvement was minimal.

We investigated this issue and found a very strong bias in the Yandex dataset, which we explain below. Most of our improvements over the RCTR baseline in the previous sections are due to queries whose responses change over time. One such query is shown in Figure 5, where the expected number of clicks at position 1 drops between days 3 and 5. If the drop was due to an unattractive item that was placed at position 1, the drop could be predicted if that item had been unattractive in the production set before. Also note that the expected number of clicks at all positions does not drop between days 3 and 5. Therefore, the RCTR estimator performs well at all positions. This changes when the positions are weighted, and we outperform it at $K = 2$, $K = 3$, and with the DCG.

The above study shows that interesting dynamics in data are necessary to outperform naive baselines. This is not surprising. If the expected number of clicks in the production and evaluation sets is similar, even simple estimators become strong baselines and we do not expect to outperform them.

Note that our estimators are data-driven, and require an overlap in the production and evaluated policies. Therefore, our approach is unsuitable for previously unseen queries. We also do not expect to perform well on infrequent queries.

7 RELATED WORK

The work described in the current paper is at the intersection of two areas. Reliable and efficient offline evaluation has been studied extensively in the bandit context, which we describe first. Then we discuss the prior work on click models, which are the starting point of our estimators, and have been shown to be representative of user behavior in various scenarios.

The problem of offline evaluation in the contextual bandit setting was first studied by Langford *et al.* [21] who provided an estimator for a stationary policy. This estimator used a variant of importance sampling and assumed that the policy does not depend on context. Many papers [10, 22, 24, 28] followed by relaxing the assumptions of this work, improving robustness, and reducing the variance of offline evaluations. None but two considered the structure of lists.

Swaminathan *et al.* [29] studied a similar click model to the IP model (Section 3.2) but with bandit feedback. In the context of lists, bandit feedback can be viewed as the total number of clicks on a list. Semi-bandit feedback, which we consider, are the indicators of clicks on each displayed item. The latter model of feedback is more informative, and Swaminathan *et al.* [29] showed in their appendix that it can lead to better results, though they did not analyze their IP estimator in detail. We study the theoretical properties of several structured estimators and evaluated them at scale.

Early work in click-based evaluation in information retrieval [18, 26] showed that higher ranks are more likely to be viewed and examined by users, and thus more likely to be clicked. Understanding and modeling of user behavior allows us to have evaluation methods that are more tolerant to the noise in behavioral data. Hofmann *et al.* [16] conducted a comprehensive survey on efficient and

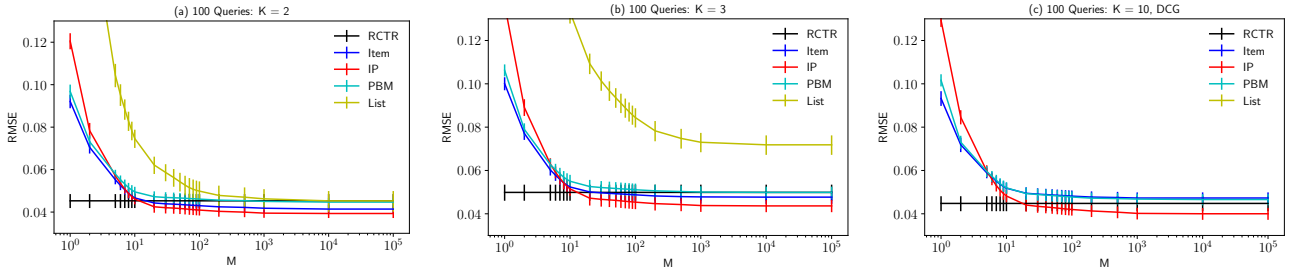

Figure 2: Prediction errors on 100 most frequent queries as a function of clipping parameter M.

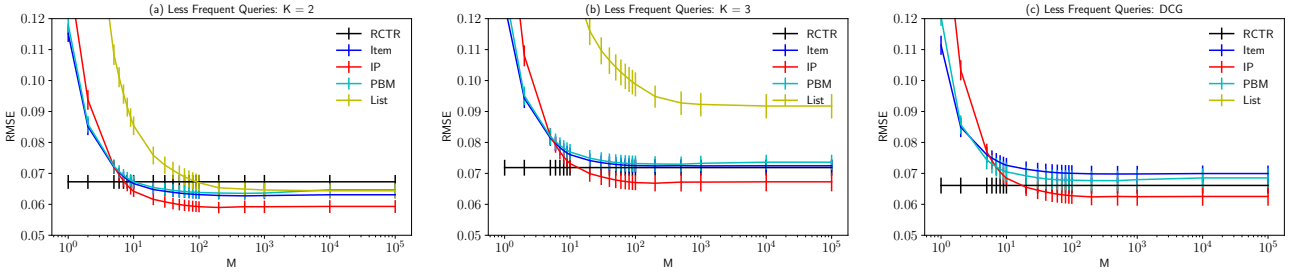

Figure 3: Prediction errors on less frequent queries as a function of clipping parameter M.

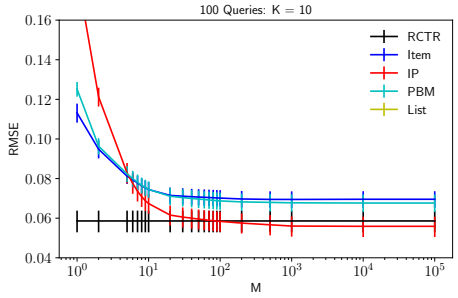

Figure 4: The errors in predicting the expected number of clicks at all $K = 10$ positions on 100 most frequent queries, as a function of clipping parameter M.

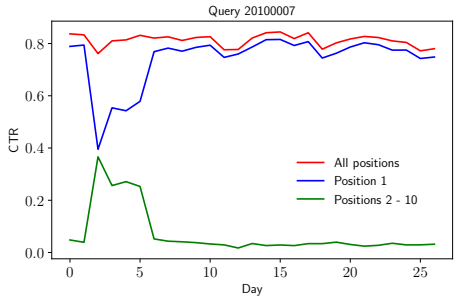

Figure 5: Expected number of clicks in query 20100007 as a function of time, in days.

reliable online evaluation of ranked lists. We focus on the offline evaluation aspect using click models.

Numerous click models have been proposed [5, 9, 12, 15], including those that we introduce in Section 3, and some models are comprehensive enough to explain finer details of user behavior. A generative model of clicks allows the evaluation of candidate ranking policies, and therefore reduce the dependence on expensive editorial judgments [11]. Hofmann *et al.* [17] used a similar importance sampling driven method to leverage historical comparisons of ranking policies to predict the outcomes of future comparisons. Click models usually have latent variables. Therefore, their parameters are estimated with an iterative EM-like procedure that lacks theoretical guarantees. In this work, we do not explicitly fit a click model. We only use the structure of the click model to represent the same independence assumptions as those in that model.

The most relevant related area to this paper is unbiased learning-to-rank and evaluation from logged data. Joachims *et al.* [19] use the sum of ranks of relevant items as a metric and Wang *et al.* [31] use the precision of clicked items, while we consider more general reward functions. They both focus on the PBM, which is only one instance of a broad class of click models. Our experimental results show that the PBM is not necessarily the best model for offline evaluation. Though we focus on evaluation [14], we show in Section 4.3 that our estimators can be used for policy optimization. Combining these estimators with models trained by batch offline processes for the many learning-to-rank objectives [25] is an interesting future direction.

Some works in information retrieval also consider user models to design metrics for ranked lists [3, 4, 8, 23, 30, 34]. Those papers do not consider the counterfactual imbalance between logged data and a new production policy, and have no theoretical guarantees in our setting.

8 CONCLUSIONS

We propose various estimators for the expected number of clicks on lists generated by ranking policies that leverage the structure of click models. We prove that our estimators are better than the unstructured list estimator, in the sense that they are less biased and have better guarantees for policy optimization. They also consistently outperform the list estimator in our experiments.

Our work can be extended in multiple directions. For instance, our key assumption is that the reward function $f(A, w)$ is linear in the contributions of individual items in A. Such functions cannot model many interesting non-linear metrics, such as the indicator of at least one click. Another potential direction for extending our work is to generalize it to click models with partial observations, such as the cascade model [9]. The main challenge in the cascade model is that the item may not be clicked due to more attractive higher-ranked items, not because it is unattractive. This phenomenon is not captured by any of our estimators.

Our estimators need to be evaluated better empirically. To the best of our knowledge, the Yandex dataset is the only public click dataset that is both large-scale and comprises clicks on individual items in recommended lists. Therefore, a better evaluation could not be done in this paper.

We also want to comment on the generality of our result. Since the reward function $f(A, w)$ is linear in the contributions of individual items in A and we do not make independence assumptions on the entries of $\boldsymbol{w} \sim D(\cdot \mid x)$, our work solves the problem of offline evaluation in stochastic combinatorial semi-bandits [6, 13, 20, 32]. Therefore, our methods can be used to estimate the values of paths in graphs from semi-bandit feedback, for instance.

REFERENCES

[1] Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael Jordan. 2003. An Introduction to MCMC for Machine Learning. *Machine Learning* 50 (2003), 5–43.

[2] Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research* 14, 1 (2013), 3207–3260.

[3] Ben Carterette. 2011. System effectiveness, user models, and user utility: a conceptual framework for investigation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 903–912.

[4] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 621–630.

[5] 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*.

[6] Wei Chen, Yajun Wang, and Yang Yuan. 2013. Combinatorial Multi-Armed Bandit: General Framework, Results and Applications. In *Proceedings of the 30th International Conference on Machine Learning*. 151–159.

[7] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. *Click Models for Web Search*. Morgan & Claypool. https://doi.org/10.2200/S00654ED1V01Y201507ICR043

[8] Aleksandr Chuklin, Pavel Serdyukov, and Maarten De Rijke. 2013. Click model-based information retrieval metrics. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 493–502.

[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*.

[10] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly Robust Policy Evaluation and Learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 1097–1104.

[11] Georges Dupret, Vanessa Murdock, and Benjamin Piwowarski. 2007. Web search engine evaluation using clickthrough data and a user model. In *WWW2007 workshop Query Log Analysis: Social and Technological Challenges*.

[12] Georges E. Dupret and Benjamin Piwowarski. 2008. 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*.

[13] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. 2012. Combinatorial Network Optimization with Unknown Variables: Multi-Armed Bandits with Linear Rewards and Individual Observations. *IEEE/ACM Transactions on Networking* 20, 5 (2012), 1466–1478.

[14] Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, and Simon Dollé. 2018. Offline A/B testing for Recommender Systems. *arXiv preprint arXiv:1801.07030* (2018).

[15] Fan Guo, Chao Liu, and Yi Min Wang. 2009. Efficient Multiple-click Models in Web Search. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*.

[16] Katja Hofmann, Lihong Li, and Filip Radlinski. 2016. Online Evaluation for Information Retrieval. *Foundations and Trends in Information Retrieval* 10, 1 (2016).

[17] Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. 2012. Estimating Interleaved Comparison Outcomes from Historical Click Data. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*.

[18] Thorsten Joachims, Laura 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, 2005*. ACM New York, 154–161.

[19] 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.

[20] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. 2015. Tight Regret Bounds for Stochastic Combinatorial Semi-Bandits. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*.

[21] John Langford, Alexander Strehl, and Jennifer Wortman. 2008. Exploration scavenging. In *Proceedings of the 25th international conference on Machine learning*. ACM, 528–535.

[22] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. 2011. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 297–306.

[23] Alistair Moffat and Justin Zobel. 2008. Rank-biased precision for measurement of retrieval effectiveness. *ACM Transactions on Information Systems (TOIS)* 27, 1 (2008), 2.

[24] Olivier Nicol, Jérémie Mary, and Philippe Preux. 2014. Improving offline evaluation of contextual bandit algorithms via bootstrapping techniques. In *Proceedings of the 31th International Conference on Machine Learning (ICML-2014), Beijing, China*, Vol. 32. 172–180.

[25] Filip Radlinski, Madhu Kurup, and Thorsten Joachims. 2008. How does clickthrough data reflect retrieval quality?. In *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 43–52.

[26] 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*.

[27] Dan Siroker and Pete Koomen. 2013. *A/B testing: The most powerful way to turn clicks into customers*. John Wiley & Sons.

[28] Alex Strehl, John Langford, Lihong Li, and Sham M Kakade. 2010. Learning from logged implicit exploration data. In *Advances in Neural Information Processing Systems*. 2217–2225.

[29] Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miroslav Dudík, John Langford, Damien Jose, and Imed Zitouni. 2016. Off-policy evaluation for slate recommendation. *arXiv preprint arXiv:1605.04812* (2016).

[30] Kuansan Wang, Toby Walker, and Zijian Zheng. 2009. PSkip: estimating relevance ranking quality from web search clickthrough data. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1355–1364.

[31] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position Bias Estimation for Unbiased Learning to Rank in Personal Search. (2018).

[32] Zheng Wen, Branislav Kveton, and Azin Ashkan. 2015. Efficient Learning in Large-Scale Combinatorial Semi-Bandits. In *Proceedings of the 32nd International Conference on Machine Learning*.

[33] Yandex 2013. Yandex Personalized Web Search Challenge. https://www.kaggle.com/c/yandex-personalized-web-search-challenge.

[34] Emine Yilmaz, Milad Shokouhi, Nick Craswell, and Stephen Robertson. 2010. Expected browsing utility for web search evaluation. In *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 1561–1564.